REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: FINITE VOLUME ELEMENT (FVE) DISCRETIZATION AND MULTILEVEL SOLUTION OF THE AXISYMMETRIC HEAT EQUATION	5. FUNDING NUMBERS
6. AUTHOR(S) Litaker, Eric T.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The axisymmetric heat equation, resulting from a point-source of heat applied to a metal block, is solved numerically; both iterative and multilevel solutions are computed in order to compare the two processes. The continuum problem is discretized in two stages: finite differences are used to discretize the time derivatives, resulting is a fully implicit backward time-stepping scheme, and the Finite Volume Element (FVE) method is used to discretize the spatial derivatives. The application of the FVE method to a problem in cylindrical coordinates is new, and results in stencils which are analyzed extensively. Several iteration schemes are considered, including both Jacobi and Gauss-Seidel; a thorough analysis of these schemes is done, using both the spectral radii of the iteration matrices and local mode analysis. Using this discretization, a Gauss-Seidel relaxation scheme is used to solve the heat equation iteratively. A multilevel solution process is then constructed, including the development of intergrid transfer and coarse grid operators. Local mode analysis is performed on the components of the amplification matrix, resulting in the two-level convergence factors for various combinations of the operators. The multilevel solution process is implemented by using multigrid V-cycles; the iterative and multilevel results are compared and discussed in detail. The computational savings resulting from the multilevel process are then discussed.

14. SUBJECT TERMS Finite Volume Element (FVE), Multigrid, Cylindrical Coordinates			15. NUMBER OF PAGES 122
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

FINITE VOLUME ELEMENT (FVE) DISCRETIZATION AND MULTILEVEL SOLUTION OF THE AXISYMMETRIC HEAT EQUATION

Eric T. Litaker
Major, United States Marine Corps
B.S. in Mathematics, University of Oklahoma, 1979
M.S. in Applied Physics, The Johns Hopkins University, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
December 1994

ABSTRACT

The axisymmetric heat equation, resulting from a point-souce of heat applied to a metal block, is solved numerically; both iterative and multilevel solutions are computed in order to compare the two processes. The continuum problem is discretized in two stages: finite differences are used to discretize the time derivatives, resulting in a fully implicit backward time-stepping scheme, and the Finite Volume Element (FVE) method is used to discretize the spatial derivatives. The application of the FVE method to a problem in cylindrical coordinates is new, and results in stencils which are analyzed extensively. Several iteration schemes are considered, including both Jacobi and Gauss-Seidel; a thorough analysis of these schemes is done, using both the spectral radii of the iteration matrices and local mode analysis. Using this discretization, a Gauss-Seidel relaxation scheme is used to solve the heat equation iteratively. A multilevel solution process is then constructed, including the development of intergrid transfer and coarse grid operators. Local mode analysis is performed on the components of the amplification matrix, resulting in the two-level convergence factors for various combinations of the operators. The multilevel solution process is implemented by using multigrid V-cycles; the iterative and multilevel results are compared and discussed in detail. The computational savings resulting from the multilevel process are then discussed.

DISCLAIMER

The computer program in Appendix B is supplied on an "as is" basis, with no warrantees of any kind. The author bears no responsibility for any consequences of using this program.

TABLE OF CONTENTS

I. INTRODUCTION

A topic of general interest is the numerical solution of partial differential equations (PDEs), such as the Navier-Stokes equation (Equation II.1). Many aspects of these types of problems present interesting challenges to constructing numerical solutions, such as non-linearities, high-speed flows that cause numerical peculiarities, unusual boundary conditions, uncommon geometries, and so on. Developing processes that solve these types of problems is difficult, and often results in complicated solutions schemes which are costly to implement. A common goal is to attempt to streamline existing solution processes, or develop new ones, in order to reduce computational complexity and cost.

At issue is how to transform a continuum problem into a discrete one (discretization), and how to construct a numerical process that will solve the discrete problem. The finite volume element (FVE) method (see [Ref. 1]) has proven to be a useful tool in developing discretizations, particularly for problems that require the enforcement of conservation laws. One of the more successful methods for streamlining the solution of PDEs, particularly elliptic equations, is a multilevel technique, to wit, multigrid (see [Ref. 2], [Ref. 3], [Ref. 1]). While this method enjoys remarkable success in solving certain classes of both linear and non-linear PDEs, its applicability to solving other types of equations awaits development. The goal of this work is twofold: to apply the FVE method to discretize a particular equation, and to implement multigrid in solving the resulting discretized problem.

The approach that is taken is to begin with a specific physical problem which results in the Navier-Stokes equation, and consider a subsidiary problem, namely the heat equation; due to the nature of the problem, cylindrical coordinates are used. The resulting problem, to which the FVE discretization and multilevel solution are applied, is the axisymmetric heat equation, involving the use of a point source. More

specifically, finite differences are used to discretize the temporal portion of the problem, resulting in the use of a fully implicit backward time-stepping scheme, and the FVE method is used to discretize the spatial part. The application of the FVE method to a problem in cylindrical coordinates is new, and results in stencils which are analyzed extensively. Once the discretization has been constructed, a number of relaxation schemes, including both Jacobi and Gauss-Seidel, are considered; a thorough analysis of these schemes is done, using both the spectral radii of the iteration matrices and local mode analysis. The goal is to choose one of the relaxation schemes for use in an iterative solution of the problem; Gauss-Seidel is the method of choice. The specifics of the multilevel technique are then developed and analyzed. Local mode analysis is performed on the components of the amplification matrix, resulting in the two-level convergence factors for various combinations of the operators. The multilevel solution process is implemented by using multigrid V-cycles; the iterative and multilevel results are compared and discussed in detail. The computational savings resulting from the multilevel process are then discussed. Insofar as multigrid is quite successful in a number of different contexts, the results of our work are somewhat disappointing in that we are not yet able to achieve the same level of success. On a more positive note, this work applies the FVE method to a problem in cylindrical coordinates, and makes use of *Maple* software to compute the necessary integrals to construct the discretization (see Chapter III and Appendix B). Additionally, the techniques that are applied do result in a solution to the problem.

II. PROBLEM STATEMENT

A. BACKGROUND

Our ultimate goal is to be able to numerically solve the Navier-Stokes equation.

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u}, \tag{II.1}$$

where $\vec{u}$ is the velocity vector, t is the time, ρ is the density, p is the pressure, and ν is the kinematic viscosity. One approach to such solutions is to consider this problem as a collection of smaller problems. By solving each of the smaller problems, we can assemble the collection of solution processes to solve the original problem. The purpose of this work is to focus on an initial step in the eventual construction of a numerical solution to Equation II.1.

One way to subdivide a problem such as this into smaller problems is to consider a specific physical problem that gives rise to the Navier-Stokes equation, and then isolate the different aspects of that problem. To that end, we consider a semi-infinite block of metal, with a heat source applied to the horizontal surface; above the horizontal surface is an (inviscid) nonconducting gas. The result is a pool of molten metal with a free surface, surrounded by the solid portion of the unmelted block (see Figure 1). The total heat flux Q is constant, and far away the solid approaches the uniform cold temperature, T_c. The resulting thermal and flow fields are assumed to be axisymmetric and steady, and are governed by conservation of energy in the solid and by conservation of energy, momentum, and mass in the pool. We therefore have the following system of equations:

$$\text{solid} \quad : \quad \frac{\partial T}{\partial t} = \kappa \nabla^2 T \tag{II.2}$$

$$\text{liquid} \quad : \quad \frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = \kappa \nabla^2 T \tag{II.3}$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} \tag{II.4}$$

$$\nabla \cdot \vec{u} = 0 \tag{II.5}$$

where T is the temperature, κ is the thermal diffusivity, and $\vec{u}$, t, ρ, p, and ν are as above.

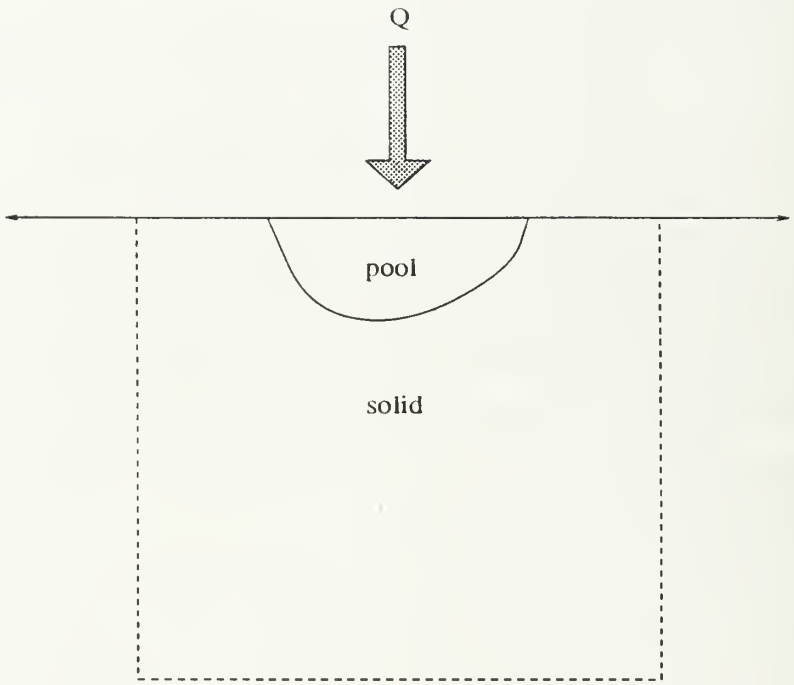

Figure 1. The molten-pool problem.

The portion of the above problem which is our focus is the conduction of heat in the solid, governed by the heat equation (Equation II.2), to which the following boundary conditions apply:

$$\text{surface} \quad z = 0 \quad : \quad k\frac{\partial T}{\partial z} = -q(r) \tag{II.6}$$

$$\text{axis} \quad r = 0 \quad : \quad \frac{\partial T}{\partial r} = 0 \tag{II.7}$$

$$\text{far away} \quad r, z \to \infty \quad : \quad T \to T_c, \tag{II.8}$$

where k is the thermal conductivity, and $q(r)$ is the imposed surface heat flux (large at $r = 0$, falling off to zero at some small value of r, such that $\int_0^\infty q(r)2\pi r dr = Q$); the condition in Equation II.7 results from the axisymmetric nature of the problem. Since the ultimate goal is to solve the Navier-Stokes equation in cylindrical coordinates, the heat equation will also be solved in this coordinate system.(see [Ref. 4])

B. POINT-SOURCE PROBLEM

Our goal is to take the first step toward solving the Navier-Stokes equation (Equation II.1) that arises from considering the molten-pool problem by solving the associated heat equation (Equation II.2). We therefore assume cylindrical geometry and azimuthal symmetry. We further simplify the conditions imposed on Equation II.2 by considering that the heat source applied to the horizontal surface of the block is a point source, with the total heat flux $Q = 1$, and that far away the solid approaches the uniform cold temperature of $T_c = 0$. That is, the imposed heat flux $q(r) = \delta(r)$ is the Dirac delta function, with $\int_0^\infty q(r) 2\pi r dr = 1$.

The existence of the solution to this type of problem is well established. However, as is the case in general when Neumann boundary conditions apply, absent a specified initial temperature distribution, the solution is not unique (see [Ref. 5]). While we are interested in solving the point-source problem numerically in cylindrical coordinates, it has spherical symmetry, so it can be solved analytically in spherical coordinates more easily. Therefore, for the analytic solution, we rewrite Equation II.2 in spherically symmetric coordinates:

$$\frac{\partial T}{\partial t} = \kappa \frac{1}{R^2} \frac{\partial}{\partial R} \left(R^2 \frac{\partial T}{\partial R} \right), \tag{II.9}$$

where R is in spherical coordinates, $R = \sqrt{r^2 + z^2}$. Beginning with an initial temperature distribution of $T = 0$ everywhere and $Q = 1$, the solution to (II.9) with boundary conditions

$$\text{surface} \quad z = 0 \quad : \quad k \frac{\partial T}{\partial z} = -q(r) = -\delta(r) \tag{II.10}$$

$$\text{axis} \quad r = 0 \quad : \quad \frac{\partial T}{\partial r} = 0 \tag{II.11}$$

$$\text{far away} \quad r, z \to \infty \quad : \quad T \to 0, \tag{II.12}$$

is found to be

$$T(R,t) = \frac{1}{2\pi\kappa R} \operatorname{erfc}\left(\frac{\mathrm{R}}{2\sqrt{\kappa \mathrm{t}}} \right), \tag{II.13}$$

where erfc(x) is the complementary error function.

$$\mathrm{erfc(x)} = 1 - \mathrm{erf(x)}, \quad \mathrm{erf(x)} = \frac{2}{\sqrt{\pi}} \int_0^{\mathrm{x}} \mathrm{e}^{-s^2} \mathrm{ds}.$$

As $t \to \infty$, the steady solution becomes

$$T(R) = \frac{1}{2\pi\kappa R} \tag{II.14}$$

(for a derivation of this result, see [Ref. 5]). In (r, z) coordinates, where $R = \sqrt{r^2 + z^2}$, these become

$$T(r, z, t) = \frac{1}{2\pi\kappa\sqrt{r^2 + z^2}} \mathrm{erfc}\left(\frac{\sqrt{\mathrm{r}^2 + \mathrm{z}^2}}{2\sqrt{\kappa \mathrm{t}}}\right), \quad \text{or} \tag{II.15}$$

$$T(r, z) = \frac{1}{2\pi\kappa\sqrt{r^2 + z^2}}. \tag{II.16}$$

For computational purposes, the idealized problem of a semi-infinite solid is truncated to a finite domain in cylindrical coordinates with azimuthal symmetry. At the far boundaries on this finite domain, homogeneous Dirichlet conditions are imposed to approximate the conditions in the unbounded problem. To further simplify computational work, we assume that the (r, z) domain is the unit grid $\Omega = [0, 1]\mathrm{x}[0, 1] \in \mathcal{R}^2$;we will eventually assume $\kappa = 1$. Thus, the equation we wish to solve is

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T, \tag{II.17}$$

with boundary conditions

$$\text{surface} \quad z = 0 \quad : \quad k\frac{\partial T}{\partial z} = -\delta(r) \tag{II.18}$$

$$\text{axis} \quad r = 0 \quad : \quad \frac{\partial T}{\partial r} = 0 \tag{II.19}$$

$$\text{far boundaries } r, z = 1 \quad : \quad T = 0. \tag{II.20}$$

III. DISCRETIZATION

In order to solve the conduction problem numerically (Equation II.2. with boundary conditions Equations II.18. II.19. and II.20). the equation representing the continuum problem must be discretized. That is. the values of the unknown function T are to be determined from a set of equations which in *some sense* approximate Equation II.2. Ultimately, a discrete approximation to the heat equation should meet the following criteria:

1) Provide for a unique solution to the problem.

2) The solution should be "close" to the exact solution for "sufficiently small" grid spacings.

3) The solution should be "effectively computable."

The significance of property 1) is obvious; property 2) relates to the questions of convergence and consistency for the discretization scheme. Property 3) relates to: a) the amount of computational work required to solve the problem, and b) the behavior of roundoff errors in the computed solution. The growth of the roundoff error is related to the notion of the stability of the discretization scheme. (see [Ref. 6])

Solution of the heat equation (Equation II.2) requires treatment of both space and time derivatives. The Finite Volume Element (FVE) method is used to discretize the spatial portion of the problem; finite differences are used to discretize the temporal portion. The approximation properties of the FVE method are fundamentally different from those associated with the finite difference method. Hence, this approach treats time and space differently. The goal in applying the FVE method to produce spatial discretization is to make use of the advantages of the Finite Volume (FV) method, its ability to be faithful to the physics in general and conservation in particular, coupled with the flexibility of the finite element representation of the unknown functions (see [Ref. 1]). As outlined in [Ref. 1], the basic approach is

to partition the spatial grid in two ways: as the union of a set of finite elements, the vertices of which comprise the grid points on which the unknowns are defined (see Figure 2), and as the union of a set of control volumes (see Figure 3), one for each grid point (the boundary points require separate treatment, as will be discussed later). The elements are used to form basis functions, a linear combination of which is used to approximate the unknown function. Upon substitution of the approximation into the equation to be solved, integrals over each control volume are taken. The integrals enforce conservation on each control volume, and therefore the partition enforces conservation on the entire domain. The system of equations generated by integration yields the discretization of the problem.

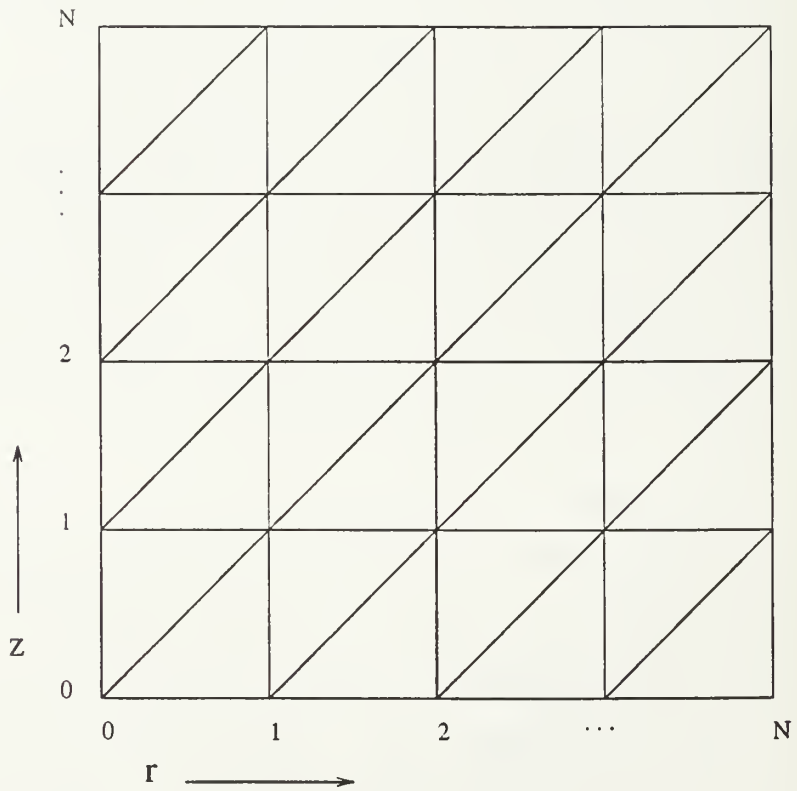

Figure 2. Partitioning of problem domain cross-section Ω by elements.

The process of discretizing Equation II.2 in both space and time is complicated, the application of the FVE method to the space derivatives being the more complex portion of this process. Therefore, before proceeding to a discussion of the FVE

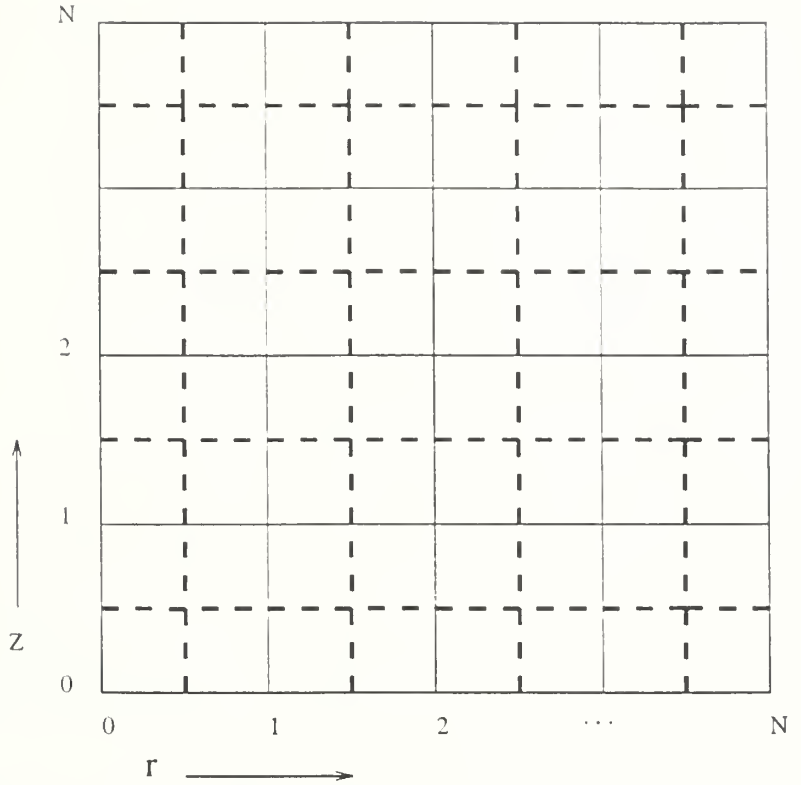

Figure 3. Partitioning of problem domain cross-section Ω into sub-volumes.

method, the finite difference method is applied to the time derivative. In order to facilitate this presentation, we present a one-dimensional example of this type of discretization (see [Ref. 7]). Consider the one-dimensional version of Equation II.2,

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}, \tag{III.1}$$

where $\kappa = 1$. One finite-difference approximation to Equation III.1 is

$$\frac{T_{k,n+1} - T_{k,n}}{g} = \frac{T_{k+1,n} - 2T_{k,n} + T_{k-1,n}}{h^2},$$

where T is the exact solution of the approximating difference equations,

$$x_k = kh \text{ and } t_n = gn, \quad (k, n = 0, 1, 2, \cdots).$$

Let $r = \frac{g}{h^2}$, and this can be written as

$$T_{k,n+1} = rT_{k-1,n} - (1 - 2r)T_{k,n} + rT_{k+1,n}.$$

Thus, we can compute the unknown $T_{k,n+1}$ at the $(n+1)$st time step using the known values from the nth time step, giving an explicit formula for determing the unknowns.[Ref. 7]

While this explicit relation provides a simple means to compute unknown values, is has a serious drawback. The process is only valid for $0 < \frac{g}{h^2} \leq \frac{1}{2}$, or $g \leq \frac{h^2}{2}$, and therefore the time step $\Delta t = g$ is necessarily small. Moreover, in order to achieve reasonable accuracy, $\Delta x = h$ must also be kept small. There is, however, a method which reduces the total amount of calculation and is valid (i.e., convergent and stable) for all finite values of r, which was proposed by Crank and Nicholson (see [Ref. 7]). The technique is to consider Equation III.1 as being satisfied at the midpoint $\{kh, (n+\frac{1}{2})g\}$ and replace $\frac{\partial^2 T}{\partial x^2}$ by the average of its finite-difference approximations at the nth and $(n+1)$st time steps. That is, the equation

$$\left(\frac{\partial T}{\partial t}\right)_{k,n+\frac{1}{2}} = \left(\frac{\partial^2 T}{\partial x^2}\right)_{k,n+\frac{1}{2}},$$

is approximated by

$$\frac{T_{k,n+1} - T_{k,n}}{g} = \frac{1}{2}\left\{\frac{T_{k-1,n+1} - 2T_{k,n+1} + T_{k+1,n+1}}{h^2} + \frac{T_{k-1,n} - 2T_{k,n} + T_{k+1,n}}{h^2}\right\},$$

which gives

$$-rT_{k-1,n+1} + (2+2r)T_{k,n+1} - rT_{k+1,n+1} = rT_{k-1,n} + (2-2r)T_{k,n} + rT_{k+1,n},$$

where $r = \frac{g}{h^2}$. Therefore, we now have three unknowns at the $(n+1)$st time step written in terms of three known values at the nth time step. Thus, the Crank-Nicholson method generates a set of equations which must be solved simultaneously; it is an implicit method.

Although the Crank-Nicholson method for Equation III.1 is stable for all positive values of r in the sense that all errors eventually tend to zero as n tends to infinity, large values of r, e.g., 40, can introduce oscillations in the solution (see [Ref. 7]). Therefore, a more general finite-difference approximation to Equation III.1 is

found by using a weighted average of the finite-difference approximations of $\frac{\partial^2 T}{\partial x^2}$ at the nth and $(n+1)$st time steps, which is given by

$$\frac{T_{k,n+1} - T_{k,n}}{g} = \alpha\left(\frac{T_{k-1,n+1} - 2T_{k,n+1} + T_{k+1,n+1}}{h^2}\right) + (1-\alpha)\left(\frac{T_{k-1,n} - 2T_{k,n} + T_{k+1,n}}{h^2}\right),$$

where $0 \leq \alpha \leq 1$ is possible. Note that $\alpha = 0$ gives the explicit scheme, $\alpha = \frac{1}{2}$, Crank-Nicholson, and $\alpha = 1$ a fully implicit backward time-difference method. The scheme is unconditionally valid (stable and convergent) for $\frac{1}{2} \leq \alpha \leq 1$, but for $0 \leq \alpha \leq \frac{1}{2}$ we must have $r \leq 1/(2-4\alpha)$. Thus, for example, for $\alpha = 0$, $r = \frac{g}{h^2} \leq \frac{1}{2}$, or $g \leq \frac{h^2}{2}$, is a requirement to guarantee validity.[Ref. 7]

We now apply this finite-difference method of discretizing the time derivative to Equation II.2. Following the work done above, a general finite-difference approximation for the time derivative of

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T \tag{III.2}$$

is given by

$$\frac{T^{n+1} - T^n}{g} = \alpha\kappa\nabla^2 T^{n+1} + (1-\alpha)\kappa\nabla^2 T^n, \tag{III.3}$$

where g is the time step, and $0 \leq \alpha \leq 1$ allows for a weighted average of the current and future time steps. The current time step is designated by n, $n+1$ is the next time step; the value of α determines the type of time stepping. However, we have determined that in order to guarantee the validity (stability and convergence) of the explicit version of this scheme ($\alpha = 0$), we must have $g \leq \frac{h^2}{2}$. We consider this value of g to be too small to be of computational use (which is verified by numerical experiments), and therefore we will use the values $\alpha = \frac{1}{2}$ and $\alpha = 1$. Thus, we have the semi-discrete relationship between the current and subsequent time steps:

$$(\frac{1}{g} - \alpha\kappa\nabla^2)T^{n+1} = (\frac{1}{g} + (1-\alpha)\kappa\nabla^2)T^n. \tag{III.4}$$

This relationship is used to establish a discrete set of equations, which gives rise to the discretization of the problem.

A. BASIS FUNCTIONS: FINITE ELEMENTS

We begin the discussion of discretizing Equation II.2 in the spatial variables by recognizing that it is necessary to make an assumption about what types of functions may be used to approximate the unknown function T; we consider that T can be approximated by continuous, piecewise linear functions. A **basis** is then chosen for the space in which these functions are found. That is, a set of functions is selected whose linear combinations determine the space of functions of interest. Therefore, we choose basis functions, $\phi_{k,j}(r,z)$, which are assumed to be continuous and piecewise linear, with

$$T(r,z) = \sum_{k=0}^{N} \sum_{j=0}^{N} \beta_{k,j} \phi_{k,j}(r,z). \tag{III.5}$$

The next step is to determine how to construct these basis functions; the element partition of the domain Ω is used as a framework for this purpose. (see Figure 2).

Since we are assuming cylindrical geometry with azimuthal symmetry, the directions of interest are r and z; a uniform grid of step size $h = \frac{1}{N}$ (N = number of grid points) is applied in both directions. In this way, we can designate function values on the grid Ω by

$$T_{k,j} = T(r_j, z_k),$$

where $z_k = kh$, $r_j = jh$, and k, j are the row and column indices. Each square is subdivided into two triangular elements by a diagonal (oriented in the direction of increasing $r + z$)[1]; the basis functions to be used in approximating the unknown function are constructed using these triangular elements. For each interior grid point, or node[2], there are six associated triangular elements: NNE, ENE, SE, SSW,WSW, NW (see Figure 4).

---

[1]The orientation of the elements is based on the necessity to apply this technique to solving the molten-pool problem. In particular, we will be required to track the movement of the phase-change boundary between solid and liquid; element orientation has been chosen to facilitate keeping track of this moving boundary.

[2]For a two-dimensional grid with N intervals in each dimension, there are $(N-1)^2$ interior grid points, $(N+1)^2$ total nodes.

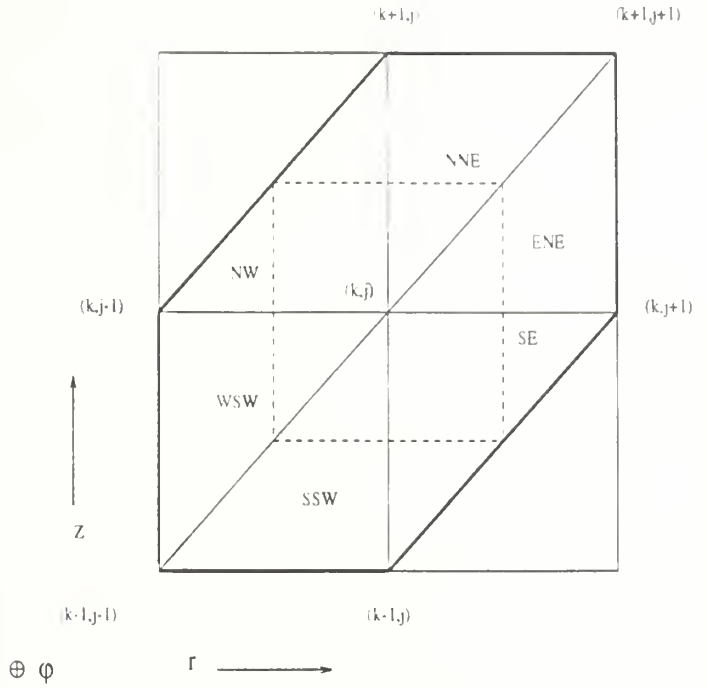

Figure 4. Hat function, top view. k is the row index (z direction), j is the column index (r direction), φ direction is into the plane of the paper ($\oplus$).

If we suppose that this collection of elements is joined along shared borders, and that the elements can be "stretched", then the finite element for a particular node, say (k, j), is formed by anchoring the collection along its external boundary (in the (r, z) plane) and extruding the grid point in a direction perpendicular to the (r, z) plane and parallel to the tangent to the φ direction at the grid point (k, j), forming the familiar "hat" function (see Figure 5). Note that

$$\phi_{k,j} = \begin{cases} 1 & \text{at the node } (r_j, z_k) \\ 0 & \text{at all other nodes} \end{cases}$$

and is piecewise linear over each triangular element, giving the hat shape.

By choosing these hat functions to be our basis functions, the coefficients of the $\phi_{k,j}(r, z)$ in Equation III.5 become the values $\beta_{k,j} = T_{k,j}$. That is, we can now specify the nodal values of T on the grid as $T_{k,j} = T(r_j, z_k)$, with

$$T(r, z) = \sum_{k=0}^{N} \sum_{j=0}^{N} T_{k,j} \phi_{k,j}(r, z). \tag{III.6}$$

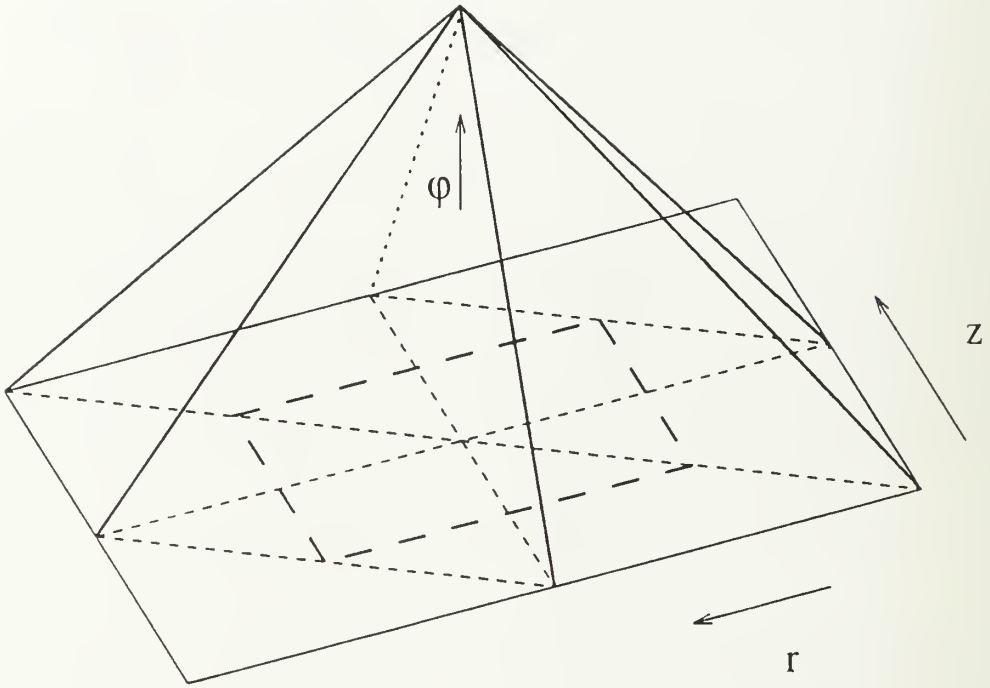

Figure 5. Hat function, oblique view.

Since the $\phi_{k,j}$ are piecewise linear, a continuous representation for function values of T is found by linear interpolation of nodal values between nodes. For example,

$$T_{k,j+\frac{1}{2}} = \frac{T_{k,j} + T_{k,j+1}}{2}.$$

Up to now, we have considered the sampled values of the unknown T at the $(N+1)^2$ grid points as an $(N+1)$x$(N+1)$ two-dimensional array; it will later be convenient to consider T as a one dimensional array. One simple method to accomplish this is to order the elements $T_{k,j}$ lexicographically. For example,

$$\begin{pmatrix} T_{0,0} & T_{0,1} & \cdots & T_{0,N+1} \\ T_{1,0} & T_{1,1} & \cdots & T_{1,N+1} \\ \vdots & \vdots & \vdots & \vdots \\ T_{N+1,0} & T_{N+1,1} & \cdots & T_{N+1,N+1} \end{pmatrix}$$

can be written as

$$(T_{0,0}, T_{0,1}, \ldots, T_{0,N+1}, T_{1,0}, T_{1,1}, \ldots, T_{1,N+1}, \ldots, T_{N+1,0}, T_{N+1,1}, \ldots, T_{N+1,N+1})^T,$$

which we will normally consider to be a column vector. Thus, we can relabel the values $\Gamma_{k,j} = \Gamma_l$, $l = (N+1)k + j + 1$ and $k, j = 0 : N+1$, in Equation III.6 so that

$$T(r,z) = \sum_{l=1}^{(N+1)^2} \Gamma_l \phi_l(r,z). \tag{III.7}$$

B. FVE STENCILS

With the construction of the basis functions over the finite elements, we can now incorporate the control volumes to complete the finite volume element discretization. Below is an example the FVE method applied to a simpler problem; the purpose is to motivate and explain the work that is necessary to handle the specifics of applying the FVE method to Equation II.2. This explanation closely follows the example in [Ref. 1].

1. Example: 2−D Potential Flow

The example problem we consider is the potential flow problem in Euclidean two-dimensional coordinates. The domain for the problem is the unit grid $\Omega = [0,1]\text{x}[0,1] \in \mathcal{R}^2$, where $x = 0$ and $y = 0$ are the near boundaries and $x = 1$ and $y = 1$ are the far boundaries. The governing equation is

$$\nabla \cdot (\rho \nabla \psi) = \eta, \tag{III.8}$$

where ρ is the density, ψ is the flow potential (i.e., ψ_x and ψ_y are the components of the fluid velocity in the x and y directions), and η is the interior source flow rate. The boundary conditions are

$$\begin{aligned} \text{near} \quad \text{boundaries} \quad &: \quad (\rho \nabla \psi) \cdot \hat{n} = \psi_1 \text{ (Neumann boundary condition)}, \\ \text{far} \quad \text{boundaries} \quad &: \quad \psi = \psi_0 \text{ (Dirichlet boundary condition)}, \end{aligned}$$

where $\hat{n}$ is the outward normal. Suppose that V is one of the control volumes in Ω, similar to those depicted in Figure 3, where r and z are replaced by x and y. Since this problem is in two dimensions, the control volumes V are actually areas, and the

corresponding surfaces S are the perimeters of these areas. Integrating Equation III.8 over V, we have

$$\int_V \nabla \cdot (\rho \nabla \psi) dV = \int_V \eta dV.$$

By applying the Gauss Divergence Theorem, the volume integral on the left-hand side becomes a surface integral.

$$\int_S (\rho \nabla \psi) \cdot \hat{n} dS = \int_V \eta dV. \tag{III.9}$$

In terms of the physical units associated with the potential flow, the relationship is

$$\frac{\text{mass}}{\text{volume}} \cdot \frac{\text{length}}{\text{time}} \cdot \text{area} = \frac{\text{mass}}{\text{time} \cdot \text{volume}} \cdot \text{volume};$$

each side represents a flow rate of mass per unit time, and $(\rho \nabla \psi) \cdot \hat{n}$ represents a flux across S. Therefore, the net flow from the interior source in V is balanced by the flow across the surface S, which can be interpreted as a mass conservation law for the control volume V.[Ref. 1]

By partitioning the domain Ω into a finite collection of control volumes (see Figure 3, where r and z are replaced by x and y), we can impose on each control volume the condition represented in Equation III.9 (the boundaries require special treatment; a discussion of these considerations is included later). As noted earlier, the imposition of conservation on each control volume results in conservation over the entire domain. This integral condition on each control volume will be used to compute the discretization; the number of discrete equations is the number of control volumes, say m, used to partition Ω. To complete the discretization process, the unknown function ψ will be approximated by β in $\mathcal{R}^m$. By appropriately choosing basis functions, we can construct an approximation v expressed in terms of its nodal values. Consider a triangular element partition similar to that in Figure 2 (where r and z are replaced by x and y) and let Υ be the space of continuous, piecewise linear functions on Ω associated with this partition. Ignoring for the moment conditions at the boundaries, the FVE discretization of Equation III.8 comes from finding a $v \in \Upsilon$

such that Equation III.9 holds for each of the control volumes in the partition of Ω. The problem in R^m is then defined when v is expressed in terms of nodal basis for Υ:

$$v = \sum_{w=1}^{m} u_w \phi_w, \tag{III.10}$$

where u_w is the value of v at the wth node and ϕ_w is the "hat function" associated with the wth node (as above, we lexicographically order the nodes, resulting in a single subscript). Substituting Equation III.10 into Equation III.9, we have the matrix equation

$$Lu = f, \tag{III.11}$$

where L is the nxn matrix with entries

$$L_{w,z} = \int_{S_w} (\rho \nabla \phi_z) \cdot \hat{n} dS \tag{III.12}$$

and

$$f_w = \int_{V_w} \eta dV \tag{III.13}$$

(except for boundary conditions).[Ref. 1]

The treatment of the boundaries depends the type of boundary condition. Neumann conditions can be imposed indirectly by substituting the flux value ψ_1 into the appropriate term in Equation III.9. Specifically, suppose we choose the control volume V that includes the origin (lower left-hand corner, Figure 3), where the surface of the control volume coincides with the boundary along the two axes. The integral condition for V is

$$\int_{x,y=0} \psi_1 dS + \int_{x,y\neq 0} (\rho \nabla \psi) \cdot \hat{n} dS = \int_V \eta dV; \tag{III.14}$$

Equation III.14 is substituted for the interior condition in Equation III.9 into the discrete approximation v for the corner volume containing the origin.[Ref. 1]

Dirichlet conditions are imposed directly; u_w takes on the value of ψ_1 at each Dirichlet node, i.e., each node on the far boundaries $(x, y = 1)$. This approach may lead to fewer unknowns u_w than equations, a problem easily resolved by discarding the

equations associated with the Dirichlet nodes. In this case, the collection of control volumes no longer partitions the domain and, therefore, conservation no longer is strictly enforced. However, it is at the Dirichlet nodes where the loss of conservation is not a concern in general.[Ref. 1]

Assuming that there is a balance in the number of equations and unknowns, we now must implement a numerical rule for evaluating the integrals in Equations III.12 and III.13. For Equation III.12, we use the following rule on each segment A that is part of the interior surface S associated with a volume V:

$$\int_A (\rho \nabla \phi) \cdot \hat{n} dS \simeq \rho(P_A) \int_A \nabla \phi \cdot \hat{n} dA$$

where P_A is point of intersection of A and the grid lines passing through the node associated with V. For Neumann boundary segments, we use the quadrature rule

$$\int_A \psi_1 dS \simeq \psi_1(P_A)|A|,$$

where $|A|$ is the "surface area", i.e., length, of A. For Equation III.13, we use

$$\int_V \eta dV \simeq \eta(N_V)|V|,$$

where N_V is the node associated with V and $|V| = \int_V dV$ is the "volume", i.e., area, of V.[Ref. 1]

Numerically evaluating Equations III.12 and III.13 yields the FVE discretization of Equation III.8. The value associated with each node, say (p, q), is a sum

$$\sum_{w=p-1,z=q-1}^{w=p+1,z=q+1} \alpha_{w,z} u_{w,z},$$

where the coefficients $\alpha_{w,z}$ are determined by the integration. A convenient way to represent the sum associated with each grid point is to place the coefficients $\alpha_{w,z}$ into a stencil,

$$\begin{bmatrix} & \alpha_{p+1,q} & \\ \alpha_{p,q-1} & \alpha_{p,q} & \alpha_{p,q+1} \\ & \alpha_{p-1,q} & \end{bmatrix} u_{p,q} = \alpha_{p-1,q} u_{p-1,q} + \alpha_{p,q-1} u_{p,q-1}$$

$$+ \alpha_{p,q} u_{p,q} + \alpha_{p+1,q} u_{p+1,q} + \alpha_{p,q+1} u_{p,q+1}$$

where, for this example, the corner values in the stencil are all zero. The value for node (p, q) is determined by applying the stencil to $u_{p,q}$.

To see what type of stencils are produced, consider the discretization on a uniform mesh with grid size $h = \frac{1}{m}$ in both coordinates. We use double subscripts $p, q = 0 : m-1$, where 0 corresponds to the Neumann boundary nodes and $m-1$ corresponds to the Dirichlet boundary nodes. Written in stencil form, the interior nodes, $0 < p, q < m-1$, for the equations in Equation III.11 are as follows ($\sum$ indicates the sum of the outer stencil entries):

$$\begin{bmatrix} & \rho((p+\frac{1}{2})h, qh) & \\ \rho(ph, (q-\frac{1}{2})h) & -\sum & \rho(ph, (q+\frac{1}{2})h) \\ & \rho((p-\frac{1}{2})h, qh) & \end{bmatrix} u_{p,q} = h^2\eta(ph, qh).$$

For the corner Neumann node $(p, q = 0)$, the stencil is given by

$$\begin{bmatrix} \frac{1}{2}\rho(\frac{h}{2}, 0) & \\ -\sum & \frac{1}{2}\rho(0, \frac{h}{2}) \end{bmatrix} u_{0,0} = \frac{h^2}{4}\eta(0,0) - \frac{h}{2}\left(\psi_1(0, \frac{h}{4}) + \psi_1(\frac{h}{4}, 0)\right).$$

For both stencils, the coefficient of the unknown to which the stencil is applied is $-\sum$.

For grid points adjacent to a Dirichlet node, the stencil entry reaching to the Dirichlet node value is moved to the right-hand side as a coefficient of the boundary data. Otherwise, the equations for these points are the same as above. For example, at the point $(0 < p < m-1, q = m-1)$ we have

$$\begin{bmatrix} & \rho((p+\frac{1}{2})h, 1-h) & \\ \rho(ph, (q-\frac{3}{2})h) & -\hat{\sum} & \\ & \rho((p-\frac{1}{2})h, 1-h) & \end{bmatrix} u_{p,m-1} = h^2\eta(ph, 1-h) - \rho(ph, 1-\frac{h}{2})\psi_0(ph, 1),$$

where $\hat{\sum}$ is the sum of the outer stencil entries without the boundary terms removed,

$$\hat{\sum} = \rho(ph, 1-\frac{h}{2}) + \rho(ph, (q-\frac{3}{2})h) + \rho((p+\frac{1}{2})h, 1-h) + \rho((p-\frac{1}{2})h, 1-h).$$

(see [Ref. 1])

2. Application to Axisymmetric Heat Equation

This technique is now used to compute the FVE stencils for Equation II.2, by combining the finite elements (see Figure 2) with the control volumes (see Figure 3). A control volume for the lth grid point, call it V_l, is a toroidal prism (see Figure 6). It is generated by taking the two-dimensional sub-volume for that point (in the (r, z) plane), and sweeping through 360^o in the φ direction.

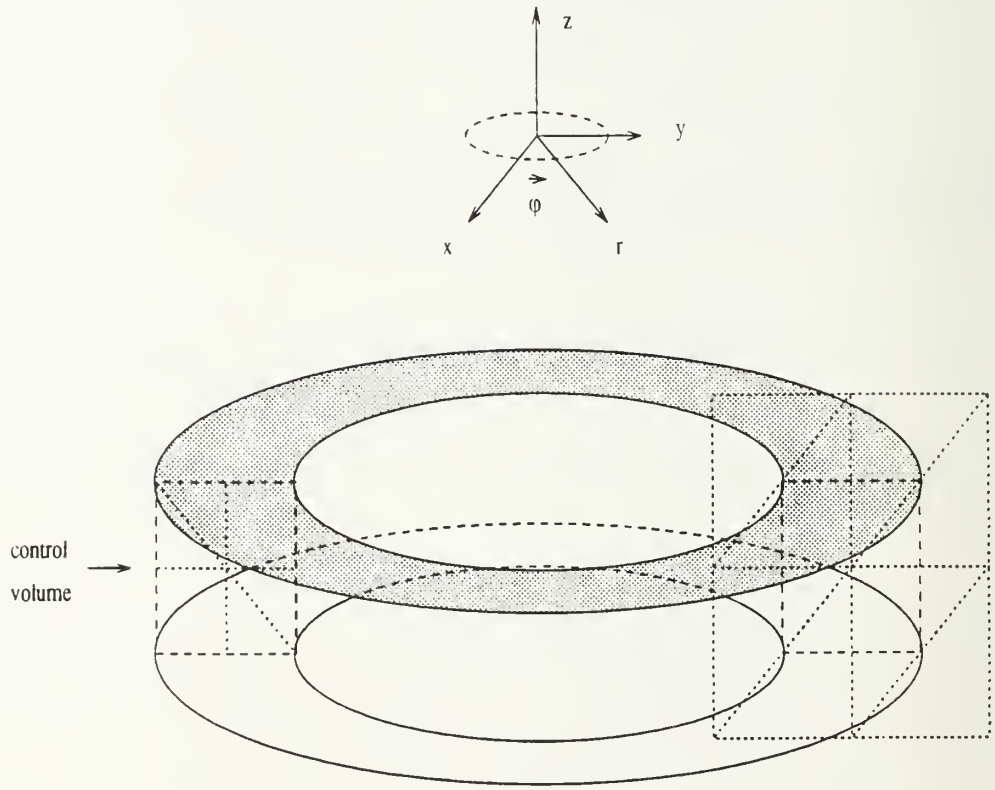

Figure 6. Toroidal volume for the conduction problem.

Integrating Equation III.4 over all control volumes V_l, where $V = \bigcup_1^{(N+1)^2} V_l$ partitions the domain Ω, we have

$$\int_V \left(\frac{1}{g} - \alpha\kappa\nabla^2\right) T^{n+1} dV = \int_V \left(\frac{1}{g} + (1-\alpha)\kappa\nabla^2\right) T^n dV, \qquad \text{(III.15)}$$

or, upon application of the Gauss Divergence Theorem the volume integral of the $\nabla^2 T$ term becomes a surface integral, and we have

$$\frac{1}{g}\int_V T^{n+1}dV - \alpha\kappa\int_S \nabla T^{n+1}\cdot\hat{n}dS = \frac{1}{g}\int_V T^n dV + (1-\alpha)\kappa\int_S \nabla T^n\cdot\hat{n}dS, \quad \text{(III.16)}$$

where $\hat{n}$ is the outward normal. Substituting the expression for approximating the unknown function T (Equation III.7), we have

$$\sum_{l=1}^{(N+1)^2}\left[\frac{1}{g}\int_V \phi_l^{n+1}dV - \alpha\kappa\int_S \nabla\phi_l^{n+1}\cdot\hat{n}dS\right]T_l^{n+1} \quad \text{(III.17)}$$
$$= \sum_{l=1}^{(N+1)^2}\left[\frac{1}{g}\int_V \phi_l^n dV + (1-\alpha)\kappa\int_S \nabla\phi_m^n\cdot\hat{n}dS\right]T_l^n,$$

where we now have integrals over each of the $(N+1)^2$ control volumes, resulting in a set of $(N+1)^2$ equations. This set of equations can be written both as an operator equation, $L[T^{n+1}] = f(T^n)$, and as a matrix equation[3], $\mathbf{M}[\vec{T}^{n+1}] = \vec{f}(\vec{T}^n)$, where the operator L and the matrix $\mathbf{M}$ are given by

$$L = \int_V\left(\frac{1}{g} - \alpha\kappa\nabla^2\right),$$
$$\mathbf{M} = \frac{1}{g}\int_V \phi_l^{n+1}dV - \alpha\kappa\int_S \nabla\phi_l^{n+1}\cdot\hat{n}dS,$$

$f(T^n)$ and $\vec{f}(\vec{T}^n)$ are given by

$$f(T^n) = \int_V\left(\frac{1}{g} + (1-\alpha)\kappa\nabla^2\right)T^n dV,$$
$$\vec{f}(\vec{T}^n) = \sum_{l=1}^{(N+1)^2}\left[\frac{1}{g}\int_V \phi_l^n dV + (1-\alpha)\kappa\int_S \nabla\phi_l^n\cdot\hat{n}dS\right]T_l^n.$$

Any function whose coefficients satisfy the resulting set of discrete equations necessarily satisfies the conservation law over any volume made up of the union of control volumes, except possibly at the boundaries. The Neumann conditions at $r = 0$ and $z = 0$ are incorporated indirectly into T, by substituting the (zero) normal derivative

[3]The operator may be represented as a continuous operator, L; as a discrete operator, L^h; or as a matrix, $\mathbf{M}$.

value into the appropriate term in Equation III.15. The Dirichlet conditions at the far boundaries are imposed directly on T; control volumes that would usually be associated with these boundary nodes are eliminated (see Figure 7). Thus, the reduced collection of control volumes no longer partitions the grid Ω, which slightly impairs conservation in the discretization. However, since we require the temperature to fall off to zero at these points, this loss of conservation is not a concern.[Ref. 1]

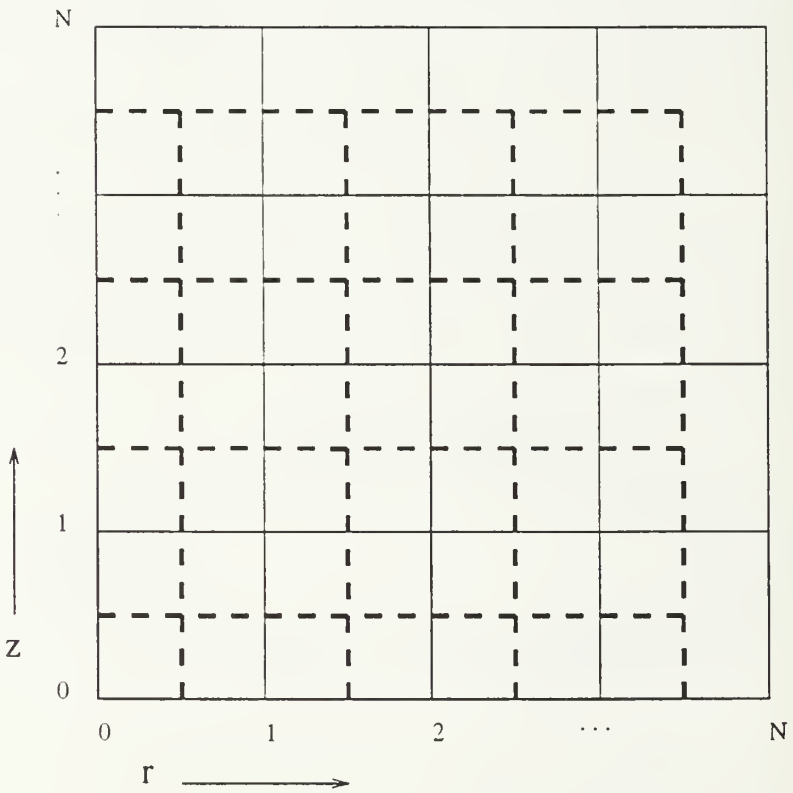

Figure 7. The problem domain cross-section is depicted after partitioning into subvolumes. Homogeneous Dirichlet boundary conditions obviate the necessity to define volumes for grid points at the far boundaries.

The integrals over the basis functions have been computed using *Maple* software and a program designed by David Canright (see Appendix B and [Ref. 8]). In order to calculate the integrals, we must be able to represent the unknown function over the basis functions, which themselves are collections of triangular planes. Therefore, the method is to determine first the function for the plane through three given

points (the cases of vertical planes and three points collinear are not considered). This procedure is then used to create the triangular elements which, when assembled, form the hat function (see Figure 5). For a given grid point, say (k, j), six of these triangular planes are joined, corresponding to the six triangles surrounding the point, NNE, ENE, SE, SSW,WSW, NW (see Figure 4). The unknown function is then interpolated over the six elements.

Once the unknown function is represented by the combination of these six interpolations, we can use *Maple* to compute the volume integrals, and the surface integrals of the gradients, in Equation III.17. The results of these integrals provide the coefficients which comprise the FVE stencils.

Thus, we have the following stencils for the volume and surface integrals respectively for interior grid points:

$$\int_{V_{k,j}} (\sum_p \sum_q T_{p,q}\phi_{p,q}) dV = \frac{h^3}{384} \begin{bmatrix} & 32j-5 & 16j+5 \\ 32j-11 & 224j & 32j+11 \\ 16j-5 & 32j+5 & \end{bmatrix} T_{k,j}, \qquad \text{(III.18)}$$

and

$$\int_{S_{k,j}} (\sum_p \sum_q \nabla T_{p,q}\phi_{k,j}) dS = \frac{h}{2} \begin{bmatrix} & 2j & \\ 2j-1 & -8j & 2j+1 \\ & 2j & \end{bmatrix} T_{k,j}, \qquad \text{(III.19)}$$

where the 2π resulting from integration in the φ direction has been factored out. (Recall that in cylindrical coordinates,

$$\int_V dV = \int_0^{2\pi} \int_{r_{j-\frac{1}{2}}}^{r_{j+\frac{1}{2}}} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} r dz dr d\varphi.)$$

Note that control volumes increase with radial distance from the origin, which is reflected in a radial bias in the stencils. More specifically, $r_j = jh$ is the radial distance from the origin, with j the column index for the unknown matrix, h the meshsize. Thus, stencil values increase with distance from the origin.

These stencils are applied to the unknown at a point on the grid, designated by its row/column position (k, j). Stencil entries indicate the values to be applied;

blank entries indicate a value of zero. Entry position indicates to which grid point the value is applied: left/right and up/down in the stencil correspond to neighboring points in those directions on the grid. That is, if stencil values are replaced by the positions to which they apply, we have

$$\begin{bmatrix} & (k+1,j) & (k+1,j+1) \\ (k,j-1) & (k,j) & (k,j+1) \\ (k-1,j-1) & (k-1,j) & \end{bmatrix}.$$

Thus, for example, the value that appears in the $(k+1,j)$ stencil position is applied to the grid point that occupies that same position.

Using techniques similar to those outlined in the two-dimensional example, boundary point stencils have been computed for the volume and surface integrals respectively. At the origin, $r = 0$, $z = 0$:

$$\frac{h^3}{384}\begin{bmatrix} & 1 & 5 \\ & 15 & 3 \end{bmatrix} T_{0,0}, \quad \text{and} \quad \frac{h}{8}\begin{bmatrix} & 1 & \\ & -3 & 2 \end{bmatrix} T_{0,0} \quad + \int q(r) r dr; \tag{III.20}$$

at $r = 0$:

$$\frac{h^3}{384}\begin{bmatrix} & 1 & 5 \\ & 25 & 11 \\ & 6 & \end{bmatrix} T_{k,0}, \quad \text{and} \quad \frac{h}{8}\begin{bmatrix} & 1 & \\ & -6 & 4 \\ & 1 & \end{bmatrix} T_{k,0}; \tag{III.21}$$

and at $z = 0$:

$$\frac{h^3}{384}\begin{bmatrix} & 32j-5 & 16j+5 \\ 24j-8 & 112j+5 & 8j+3 \end{bmatrix} T_{0,j}, \quad \text{and} \quad \frac{h}{4}\begin{bmatrix} & 4j & \\ 2j-1 & -8j & 2j+1 \end{bmatrix} T_{0,j}. \tag{III.22}$$

For points adjacent to the far boundaries, the stencils in Equations III.18, III.19, III.21, and III.22 are applied. However, since the homogeneous Dirichlet conditions dictate that these boundary values are zero, the resulting contribution after stencil application remains zero. Thus, in effect, far-boundary values do not contribute to the stencils for points adjacent to these boundaries.

We can now combine all of the above stencils to generate the operator L^h,

$$L^h[T^{n+1}] = f^h(T^n), \tag{III.23}$$

where h is the step size on the grid. The matrix representation of L^h, $\mathbf{M}$, is a block tridiagonal $(N+1)^2 \mathrm{x} (N+1)^2$ matrix of the form

$$\mathbf{M} = \begin{bmatrix} A & \Delta & 0 & 0 & \cdots & 0 & 0 \\ \Gamma & A & \Delta & 0 & \cdots & 0 & 0 \\ 0 & \Gamma & A & \Delta & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \Gamma & A & \Delta \\ 0 & \cdots & \cdots & \cdots & 0 & \Gamma & A \end{bmatrix},$$

where A, Δ, and Γ are $(N+1)\mathrm{x}(N+1)$ generic banded matrices (not all identical); A is tridiagonal, Δ is upper bidiagonal, and Γ is lower bidiagonal. When $\mathbf{M}$ multiplies the matrix of $(N+1)^2$ values of T, arranged lexicographically as a column vector, the matrices A, Δ, and Γ produce the effect of the stencils "reaching" function values respectively on the current row, the row above, and the row below. Numerical experiments using *Matlab* to construct the matrix $\mathbf{M}$ for $N = 8, 12, 16, 20$, and 24, with $g = h$ and $\alpha = \frac{1}{2}$ and 1, indicate that it has full rank. Thus, we expect a unique solution to the linear algebra problem which is a discretization of the heat equation.

IV. RELAXATION SCHEMES

The FVE method, with weighted-average time-stepping, has been used to discretize the continuum problem,

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T, \tag{IV.1}$$

by

$$L^h[T^{n+1}] = f^h(T^n) \tag{IV.2}$$

or, in matrix form,

$$\mathbf{M}[\vec{T}^{n+1}] = \vec{f}(\vec{T}^n), \tag{IV.3}$$

on a grid of meshsize $h = \frac{1}{N}$. The input for these equations is the solution at the current time step, $\vec{T}^n$ (where T^n and $\vec{T}^n$ are used interchangeably), the result of solving the equations is the solution at the next time step, $\vec{T}^{n+1}$. As was indicated in Chapter III, there is good reason to believe that the matrix $\mathbf{M}$ is of full rank and, thus, expect a unique solution to exist for the linear algebra problem that arises from the discretization of the continuum problem. Solution by direct methods requires factorization of $\mathbf{M}$ and, since $\mathbf{M}$ is both large and sparse, solution by direct methods may be impractical. We therefore consider iterative methods to solve the matrix equation at each time step. These methods generate, for each time step, a sequence of approximate solutions, $\{\vec{T}_{(s)}\}$[1]; the choice of relaxation scheme determines whether or not the sequence $\{\vec{T}_{(s)}\}$ converges. Upon implementation of a relaxation scheme that produces a converging sequence of approximate solutions at each time step, we begin with an initial temperature distribution, $\vec{T}^0$, and the solutions are stepped in time. The desired result is a sequence of solutions, corresponding to a sequence of time steps, which tends to steady state. This process allows for the evaluation of

[1]Here the subscript (s) indexes the sequence of approximate solutions; elsewhere, subscripts without parentheses indicate grid position or vector components.

various values for the time step, as well as of various weightings used in averaging the current and subsequent time steps.

A. ITERATION MATRICES

It is common in constructing iterative methods to propose a **splitting** for the matrix $M = E - F$, where linear systems of the form $E\vec{x} = \vec{b}$ are "easy" to solve (see [Ref. 9]). The sequence of approximations, $\{\vec{T}_{(s)}\}$, is generated by $E\vec{T}_{(s+1)} = F\vec{T}_{(s)} + \vec{f}$ and, therefore, it is natural to define an **iteration matrix**, $P = E^{-1}F$, so that $\vec{T}_{(s+1)} = P\vec{T}_{(s)} + E^{-1}\vec{f}$. Additionally, if $\vec{T}^*$ is the exact solution so that $M\vec{T}^* = \vec{f}$, then $E\vec{T}^* = F\vec{T}^* + \vec{f}$ and $\vec{T}^* = E^{-1}F\vec{T}^* + E^{-1}\vec{f}$. Hence the solution, $\vec{T}^*$, is a **fixed** point of the iteration $\vec{T}_{(s+1)} = E^{-1}F\vec{T}_{(s)} + E^{-1}\vec{f}$.

The matrix $P = E^{-1}F$ is also called the **error propagation matrix**, since if $\vec{T}_{(s+1)} = P\vec{T}_{(s)} + E^{-1}\vec{f}$ or $\vec{T}_{(s+1)} = P\vec{T}_{(s)} + \vec{B}$ ($\vec{B}$ a constant vector), and $\vec{e}_{(s+1)}$ is the error at the $(s+1)$st step, then

$$\begin{aligned}
\vec{e}_{(s+1)} &= \vec{T}_{(s+1)} - \vec{T}^* \\
&= [P\vec{T}_{(s)} + \vec{B}] - [P\vec{T}^* + \vec{B}] \text{ (since } \vec{T}^* = P\vec{T}^* + \vec{B}) \\
&= P\vec{T}_{(s)} - P\vec{T}^* \\
&= P(\vec{T}_{(s)} - \vec{T}^*),
\end{aligned}$$

and therefore

$$\vec{e}_{(s+1)} = P\vec{e}_{(s)}. \tag{IV.4}$$

Using induction, it is easy to show that $\vec{e}_{(s)} = P^s\vec{e}_{(0)}$. By Equation IV.4, $\vec{e}_{(1)} = P\vec{e}_{(0)}$ and

$$\begin{aligned}
\vec{e}_{(2)} &= P\vec{e}_{(1)} \\
&= P[P\vec{e}_{(0)}] \\
&= P^2\vec{e}_{(0)}.
\end{aligned}$$

By induction, assume that $\vec{e}_{(k)} = \mathbf{P}^k \vec{e}_{(0)}$, in order to show that $\vec{e}_{(k+1)} = \mathbf{P}^{(k+1)} \vec{e}_{(0)}$. Now,

$$\begin{aligned} \vec{e}_{(k+1)} &= \mathbf{P}\vec{e}_{(k)} \\ &= \mathbf{P}[\mathbf{P}^k \vec{e}_{(0)}] \text{ (by the induction hypothesis)} \\ &= \mathbf{P}^{(k+1)} \vec{e}_{(0)}. \end{aligned}$$

Thus, since $\vec{e}_{(k+1)} = \mathbf{P}^{(k+1)} \vec{e}_{(0)}$, we conclude $\vec{e}_{(s)} = \mathbf{P}^s \vec{e}_{(0)}$, for s any integer. Additionally,

$$\|\vec{e}_s\| = \|\mathbf{P}^s \vec{e}_{(0)}\| \leq \|\mathbf{P}\|^s \|\vec{e}_{(0)}\|, \tag{IV.5}$$

which will be useful later.

One of the simplest relaxation methods is the **Jacobi** method, also referred to as simultaneous displacement (for a more detailed account of all of these methods, see [Ref. 3] or [Ref. 9]). It is produced by solving the lth equation of Equation IV.3 for the lth unknown, T_l, $l = 1 : (N+1)^2$. Before proceeding to a discussion of the iteration matrix, we present the component form for this iteration scheme:

$$T_{k,j}^{(new)} \equiv T_l^{(new)} = \frac{f_l - [\frac{h^3}{g384}(\hat{\sum}_V^{old}) - \frac{\alpha\kappa h}{2}(\hat{\sum}_S^{old})]}{\frac{h^3}{g384}224j + \frac{\alpha\kappa h}{2}8j}, \tag{IV.6}$$

where $\hat{\sum}_V$ and $\hat{\sum}_S$ are, respectively, the sum of volume stencil entries and surface stencil entries applied to the current values of the unknown, $T_{k,j}^{old}$, except at the grid point on which the stencil is centered:

$$\hat{\sum}_V^{old} = \begin{pmatrix} & (32j-5)T_{k+1,j}^{(old)} & +(16j+5)T_{k+1,j+1}^{(old)} \\ +(32j-11)T_{k,j-1}^{(old)} & & +(32j+11)T_{k,j+1}^{(old)} \\ +(16j-5)T_{k-1,j-1}^{(old)} & +(32j+5)T_{k-1,j}^{(old)} & \end{pmatrix}, \tag{IV.7}$$

and

$$\hat{\sum}_S^{old} = \begin{pmatrix} & 2jT_{k+1,j}^{(old)} & \\ +(-1+2j)T_{k,j-1}^{(old)} & & +(1+2j)T_{k,j+1}^{(old)} \\ & +2jT_{k-1,j}^{(old)} & \end{pmatrix}. \tag{IV.8}$$

One iteration sweep consists of computing Equation IV.6 for each component of the unknown vector T. Provided that the process converges, the sweeps continue until a desired level of convergence is reached.

Another, more succinct, way to present this method is in matrix form. If we write the operator matrix as the sum of a diagonal matrix ($\mathbf{D}$), a strictly upper triangular matrix ($\mathbf{U}$), and a stictly lower triangular matrix ($\mathbf{L}$),

$$\mathbf{M} = \mathbf{D} + \mathbf{U} + \mathbf{L},$$

the matrix equation to be solved becomes

$$(\mathbf{D} + \mathbf{U} + \mathbf{L})[\vec{T}] = \vec{f}.$$

Now define $\mathbf{E}_J = \mathbf{D}$ and $\mathbf{F}_J = -(\mathbf{U} + \mathbf{L})$, so that this may be written as

$$\begin{aligned} \mathbf{D}[\vec{T}] &= -(\mathbf{U} + \mathbf{L})[\vec{T}] + \vec{f}, \quad \text{or} \\ \mathbf{E}_J[\vec{T}] &= \mathbf{F}_J[\vec{T}] + \vec{f}, \end{aligned}$$

or as

$$\begin{aligned} \vec{T} &= -\mathbf{D}^{-1}(\mathbf{U} + \mathbf{L})[\vec{T}] + \mathbf{D}^{-1}\vec{f}, \quad \text{or} \\ \vec{T} &= \mathbf{E}_J^{-1}\mathbf{F}_J[\vec{T}] + \mathbf{E}_J^{-1}\vec{f}, \end{aligned}$$

which corresponds to solving the lth equation for T_l, $l = 1 : (N+1)^2$. If we define the Jacobi iteration matrix as

$$\begin{aligned} \mathbf{P}_J &= \mathbf{E}_J^{-1}\mathbf{F}_J, \quad \text{or} \\ &= -\mathbf{D}^{-1}(\mathbf{U} + \mathbf{L}), \end{aligned}$$

the matrix form of the Jacobi method becomes

$$\vec{T}^{(new)} = \mathbf{P}_J[\vec{T}^{(old)}] + \mathbf{E}_J^{-1}\vec{f}. \tag{IV.9}$$

A modified version of this method, called the **weighted Jacobi** method, is determined by introducing a weight, $0 \leq \omega \leq 1$ ($\omega = 1$ is the original Jacobi method) (see [Ref. 3]). The matrix form is

$$\vec{T}^{(new)} = \mathbf{P}_\omega[\vec{T}^{(old)}] + \omega \mathbf{E}_\omega^{-1} \vec{f}. \tag{IV.10}$$

where $\mathbf{I}$ is the indentity matrix, $\mathbf{E}_\omega = \mathbf{D}$, and $\mathbf{F}_\omega = (1-\omega)\mathbf{D} - \omega(\mathbf{U}+\mathbf{L})$, and

$$\begin{aligned} \mathbf{P}_\omega &= \mathbf{E}_\omega^{-1}\mathbf{F}_\omega \\ &= \mathbf{D}^{-1}[(1-\omega)\mathbf{D} - \omega(\mathbf{U}+\mathbf{L})] \\ &= (1-\omega)\mathbf{I} + \omega \mathbf{P}_J \end{aligned}$$

is the weighted Jacobi iteration matrix. In this method, the new approximation is a weighted average of an intermediate approximation and the old approximation; the intermediate approximation is the Jacobi iterate of the old approximation.

The weighted Jacobi method computes all of the components of the new approximation before it begins to use them in the next approximation. This requires storing both the current and new approximations; a simple modification allows for updating the current approximation "in place", using only the storage required for one approximation. The **Gauss-Seidel** method incorporates the new changes as soon as they are computed by overwriting the current approximation with the new (see [Ref. 3]). More important, however, is that (on model problems) the Gauss-Seidel method generally converges about twice as fast at the Jacobi method (see [Ref. 9]). In component form, this iteration scheme is

$$T_{k,j}^{(new)} \equiv T_l^{(new)} \leftarrow \frac{f_l - [\frac{h^3}{g384}(\breve{\sum}_V) - \frac{\alpha\kappa h}{2}(\breve{\sum}_S)]}{\frac{h^3}{g384}224j + \frac{\alpha\kappa h}{2}8j}, \tag{IV.11}$$

where the arrow indicates overwriting, and

$$\breve{\sum}_V = \begin{pmatrix} & (32j-5)T_{k+1,j}^{(old)} & +(16j+5)T_{k+1,j+1}^{(old)} \\ +(32j-11)T_{k,j-1}^{(new)} & & +(32j+11)T_{k,j+1}^{(old)} \\ +(16j-5)T_{k-1,j-1}^{(new)} & +(32j+5)T_{k-1,j}^{(new)} & \end{pmatrix}, \tag{IV.12}$$

and

$$\sum_S = \begin{pmatrix} & 2jT_{k+1,j}^{(old)} & \\ +(-1+2j)T_{k,j-1}^{(new)} & & +(1+2j)T_{k,j+1}^{(old)} \\ & +2jT_{k-1,j}^{(new)} & \end{pmatrix}. \qquad \text{(IV.13)}$$

In matrix form, using $\mathbf{M} = \mathbf{D} + \mathbf{U} + \mathbf{L}$, $\mathbf{E}_G = (\mathbf{D} + \mathbf{L})$, and $\mathbf{F}_G = -\mathbf{U}$, we have

$$\begin{aligned} (\mathbf{D}+\mathbf{L})[\vec{T}] &= -\mathbf{U}[\vec{T}] + \vec{f}, \quad \text{or} \\ \mathbf{E}_G[\vec{T}] &= \mathbf{F}_G[\vec{T}] + \vec{f}, \end{aligned}$$

or

$$\begin{aligned} \vec{T}^{(new)} &\leftarrow -(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}[\vec{T}^{(old)}] + (\mathbf{D}+\mathbf{L})^{-1}\vec{f}, \quad \text{or} \\ \vec{T}^{(new)} &\leftarrow \mathbf{E}_G^{-1}\mathbf{F}_G[\vec{T}^{(old)}] + \mathbf{E}_G^{-1}\vec{f}. \end{aligned}$$

This corresponds to solving the lth equation for T_l, using the new approximations for components $1, 2, \ldots, l-1$. Now, define the Gauss-Seidel iteration matrix,

$$\begin{aligned} \mathbf{P}_G &= \mathbf{E}_G^{-1}\mathbf{F}_G \\ &= -(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}, \end{aligned}$$

so that the iteration scheme in matrix form becomes

$$\vec{T}^{(new)} \leftarrow \mathbf{P}_G[\vec{T}^{(old)}] + \mathbf{E}^{-1}\vec{f}. \qquad \text{(IV.14)}$$

With the above lexicographic ordering of the $(N+1)^2$ components of $\vec{T}$ and the components updated in ascending order, the effect of a Gauss-Seidel sweep is to start at $r = 0$ and update in the radial direction for each vertical step, starting at $z = 0$ (see Figure 8).

As with the weighted Jacobi method, we can make a modification to the Gauss-Seidel method. Define a parameter $\gamma \in \mathcal{R}$ ($\gamma = 1$ is the original Gauss-Seidel method) and the modified method is **successive over-relaxation, SOR**, (see [Ref. 9]) which, in matrix form, is given by

$$\vec{T}^{(new)} = \mathbf{P}_\gamma \vec{T}^{(old)} + \gamma \mathbf{E}_\gamma^{-1}\vec{f}, \qquad \text{(IV.15)}$$

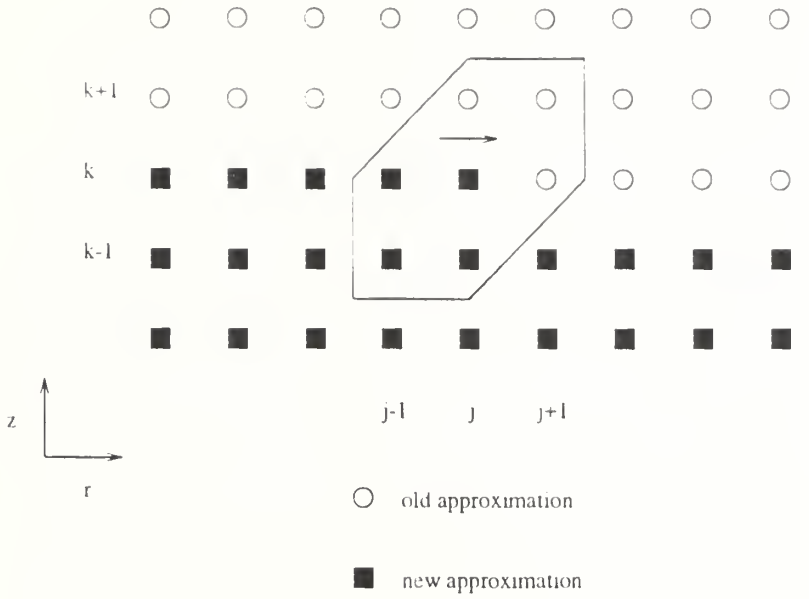

Figure 8. Gauss-Seidel sweep.

where, with $\mathbf{E}_\gamma = \mathbf{D} + \gamma\mathbf{L}$ and $\mathbf{F}_\gamma = (1-\gamma)\mathbf{D} - \gamma\mathbf{U}$, we have

$$\begin{aligned} \mathbf{P}_\gamma &= (\mathbf{D} + \gamma\mathbf{L})^{-1}[(1-\gamma)\mathbf{D} - \gamma\mathbf{U}] \\ &= \mathbf{E}_\gamma^{-1}\mathbf{F}_\gamma. \end{aligned}$$

Similar to the weighted Jacobi method, *SOR* is a weighted average of an intermediate approximation and the old approximation.

The question of interest now is whether or not the sequence of approximations, $\{\vec{T}_{(s)}\}$, generated by $\mathbf{E}\vec{T}_{(s+1)} = \mathbf{F}\vec{T}_{(s)} + \vec{f}$, converges. Therefore, we make use of the following theorem:

Theorem IV.1 *Suppose that* $\vec{f} \in \mathcal{R}^n$ *and* $\mathbf{M} = \mathbf{E} - \mathbf{F} \in \mathcal{R}^{n\times n}$ *is nonsingular. If* $\mathbf{E}$ *is nonsingular and the spectral radius of* $\mathbf{E}^{-1}\mathbf{F}$ *satisfies* $\rho(\mathbf{E}^{-1}\mathbf{F}) < 1$, *then the iterates* $\vec{T}_{(s)}$, *defined by* $\mathbf{E}\vec{T}_{(s+1)} = \mathbf{F}\vec{T}_{(s)} + \vec{f}$, *converge to* $\vec{T} = \mathbf{M}^{-1}\vec{f}$ *for any starting vector* $\vec{T}_{(0)}$.

The proof is found in [Ref. 9], and makes use of the following lemma:

Lemma IV.1 *If* $\rho(\mathbf{E}^{-1}\mathbf{F}) < 1$, *then* $(\mathbf{E}^{-1}\mathbf{F})^k \to 0$ *as* $k \to \infty$.

	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
Jacobi	.9259	.9544	.9675	.9749	.9796
$\omega = \frac{1}{3}$	.9721	.9826	.9875	.9903	.9921
$\omega = \frac{2}{3}$	.9442	.9652	.9750	.9806	.9841
Gauss-Seidel	.8395	.8982	.9263	.9425	.9530
$\gamma = \frac{1}{3}$	.9670	.9793	.9851	.9884	.9905
$\gamma = \frac{2}{3}$	.9189	.9488	.9630	.9711	.9764

Table I. Spectral Radii for Crank-Nicholson Time Stepping ($\alpha = \frac{1}{2}$).

	$N = 8$	$N = 12$	$N = 16$	$N = 20$	$N = 24$
Jacobi	.9499	.9709	.9801	.9850	.9881
$\omega = \frac{1}{3}$	.9817	.9892	.9925	.9943	.9955
$\omega = \frac{2}{3}$	.9634	.9784	.9850	.9887	.9909
Gauss-Seidel	.8931	.9362	.9556	.9663	.9730
$\gamma = \frac{1}{3}$	.9782	.9871	.9910	.9932	.9946
$\gamma = \frac{2}{3}$	.9462	.9680	.9777	.9831	.9865

Table II. Spectral Radii for Implicit Time Stepping ($\alpha = 1$).

The matrices $\mathbf{M}$, $\mathbf{E}_J$, and $\mathbf{E}_G$, and iteration matrices, $\mathbf{P}$, for various values of N, $\alpha = \frac{1}{2}$ and 1, and $g = h$, have been constructed using *Matlab*. We have experimentally verified that $\mathbf{M}$, $\mathbf{E}_J$, and $\mathbf{E}_G$ are nonsingular, and the spectral radii of the error propagation matrices, $\mathbf{P}$, have been computed. Due to memory limitations with *Matlab*, the calculations are made for relatively small values of N. The results are indicated in Tables I and II, where the value of α indicates the type of time stepping, and the values of ω and γ are the weightings in the weighted Jacobi and *SOR* methods respectively.

The results of these numerical experiments indicate that both $\rho(\mathbf{P}_J) < 1$ and $\rho(\mathbf{P}_G) < 1$, with $\rho(\mathbf{P}_G) < \rho(\mathbf{P}_J)$. The modifications to the Jacobi and Gauss-Seidel methods do not provide any improvement; the spectral radii for both methods are higher for the modified iteration matrices than for the original iteration matrices.

Thus, the Gauss-Seidel method, used with either the Crank-Nicholson or implicit time-stepping scheme, appears to be the best choice of these relaxation schemes. However, if the spectral radius of $\mathbf{P}$ is near unity, convergence may by unacceptably slow since the error tends to 0 like $\rho(\mathbf{P})^k$, as indicated by the lemma and $\|\vec{\epsilon}^s\| \leq \|\mathbf{P}\|^s \|\vec{\epsilon}_{(0)}\|$ (Equation IV.5). As is evident from the above tables, even for a moderately spaced grid (e.g., $N = 16$, $g = h$, and $\alpha = \frac{1}{2}$ or 1), $\rho(\mathbf{P}) > .9$, and the spectral radius increases with N.

B. ALTERNATIVE SCHEMES: LINE RELAXATION

Both the Jacobi and Gauss-Seidel methods give rise to iteration matrices, are implemented in a straightforward manner, and are attractive because of their simplicity. While the ease of implementation is a significant advantage, the convergence properties of either or both may not be acceptable and, therefore, alternative schemes may be desirable. One such, which seems reasonable based on the geometry of the problem, is line relaxation. There are two obvious options in this regard: radial or vertical line relaxation (see Figures 9 and 10), where either an entire row or column is updated simultaneously. Both options require the solution of an $(N+1)$x$(N+1)$ tridiagonal matrix for each row/column update in the unknown matrix. That is, while the previously outlined relaxation schemes (point relaxation) proceed by successively solving a collection of algebraic equations for one unknown, line relaxation proceeds by solving a succession of matrix equations. For example, suppose that the matrices A, Δ, and Γ are tridiagonal, upper bidiagonal, and lower bidiagonal, respectively, and that T_i and f_i are the portions of the respective unknown and right hand side vectors in Equation IV.3 that are rows in their corresponding matrices:

$$\begin{bmatrix} A & \Delta & 0 & 0 \\ \Gamma & A & \Delta & 0 \\ 0 & \Gamma & A & \Delta \\ 0 & 0 & \Gamma & A \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}.$$

Radial line relaxation consists of solving the following succession of matrix equations:

$$\begin{aligned} T_1 &\leftarrow A^{-1}(f_1 - \Delta T_2), \\ T_2 &\leftarrow A^{-1}(f_2 - \Gamma T_1 - \Delta T_3), \\ T_3 &\leftarrow A^{-1}(f_3 - \Gamma T_2 - \Delta T_4), \text{ and} \\ T_4 &\leftarrow A^{-1}(f_4 - \Gamma T_3). \end{aligned}$$

In order to illustrate how line relaxation works relative to the FVE stencils, let

$$\sum_{V}^{row} = \begin{pmatrix} & (32j-5)T_{k+1,j}^{(old)} & +(16j+5)T_{k+1,j+1}^{(old)} \\ \\ +(16j-5)T_{k-1,j-1}^{(new)} & +(32j+5)T_{k-1,j}^{(new)} & \end{pmatrix} \qquad \text{(IV.16)}$$

and

$$\sum_{S}^{row} = \begin{pmatrix} & 2jT_{k+1,j}^{(old)} & \\ \\ & +2jT_{k-1,j}^{(new)} & \end{pmatrix}, \qquad \text{(IV.17)}$$

and

$$\sum_{V}^{column} = \begin{pmatrix} & & (16j+5)T_{k+1,j+1}^{(old)} \\ +(32j-11)T_{k,j-1}^{(new)} & & +(32j+11)T_{k,j+1}^{(old)} \\ +(16j-5)T_{k-1,j-1}^{(new)} & & \end{pmatrix} \qquad \text{(IV.18)}$$

and

$$\sum_{S}^{column} = \begin{pmatrix} +(-1+2j)T_{k,j-1}^{(new)} & & +(1+2j)T_{k,j+1}^{(old)} \\ \\ \end{pmatrix}. \qquad \text{(IV.19)}$$

Now, define

$$\begin{aligned} \Upsilon_{k,j}^{(new)row} &= \frac{h^3}{g384}[(32j-11)T_{k,j-1}^{(new)} + 224jT_{k,j}^{(new)} + (32j+11)T_{k,j+1}^{(new)}] \\ &\quad -\frac{\alpha\kappa h}{2}[(-1+2j)T_{k,j-1}^{(new)} - 8jT_{k,j}^{(new)} + (1+2j)T_{k,j+1}^{(new)}], \text{ and} \end{aligned}$$

$$\Upsilon_{k,j}^{(new)column} = \frac{h^3}{g384}[(32j+5)T_{k-1,j}^{(new)} + 224jT_{k,j}^{(new)} + (32j-5)T_{k+1,j}^{(new)}]$$
$$-\frac{\alpha\kappa h}{2}[2jT_{k-1,j}^{(new)} - 8jT_{k,j}^{(new)} + 2jT_{k+1,j}^{(new)}],$$

so that, for the radial and vertical relaxations respectively, we must solve

$$\text{radial} \quad : \quad \Upsilon_{k,j}^{(new)row} \leftarrow f_l - [\frac{h^3}{g384}(\sum_{V}^{row}) - \frac{\alpha\kappa h}{2}(\sum_{S}^{row})] \quad \text{or} \qquad \text{(IV.20)}$$

$$\text{vertical} \quad : \quad \Upsilon_{k,j}^{(new)column} \leftarrow f_l - [\frac{h^3}{g384}(\sum_{V}^{column}) - \frac{\alpha\kappa h}{2}(\sum_{S}^{column})] \qquad \text{(IV.21)}$$

simultaneously for each row or column in the unknown matrix.

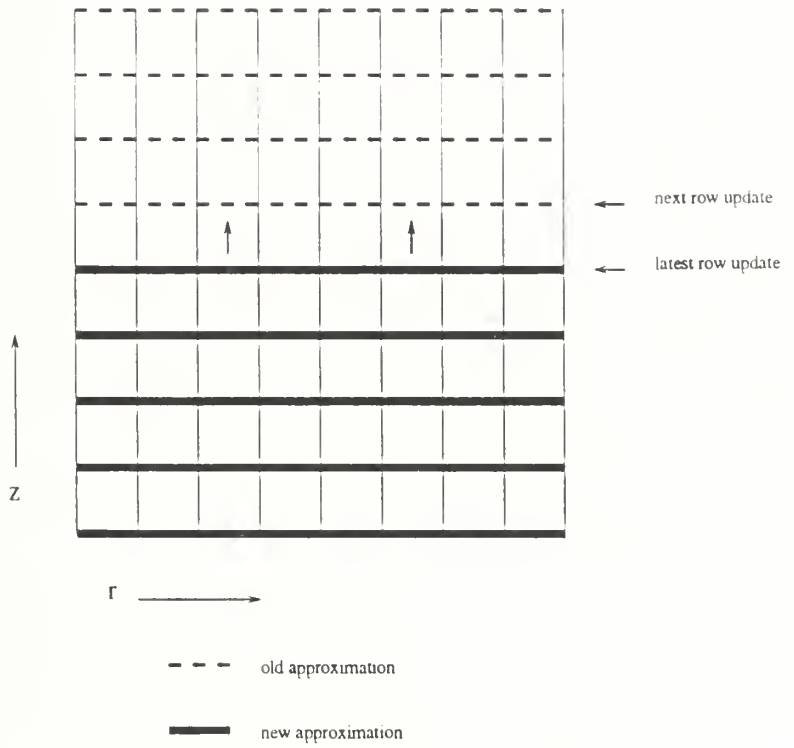

Figure 9. Updating an entire row at one time.

The computational cost of updating an entire row/column at a time is higher than a row/column update using either the Jacobi or Gauss-Seidel methods. However, this type of relaxation may be sufficiently efficacious to warrant the extra expense, in that convergence may be achieved significantly faster. For comparison, we now consider local mode analysis of the Jacobi and Gauss-Seidel methods and line relaxation.

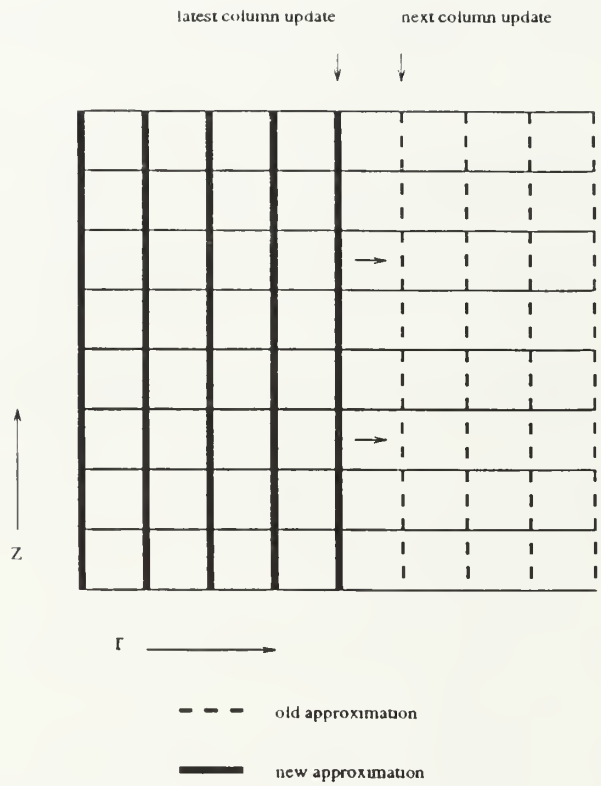

Figure 10. Updating an entire column at a time.

C. LOCAL MODE ANALYSIS

While an examination of the spectral radii of iteration (error progagation) matrices can be instructive, it is often useful to conduct a more detailed analysis. What follows is a local mode analysis of various iteration schemes. Since the error propagation matrices indicate how the error evolves during the iteration process, we use a DFT (see Appendix A, [Ref. 10]) to expand components of the error equations associated with the various relaxation schemes. The coefficients in these expansions are the factors by which the corresponding modes of the error are magnified/reduced for each relaxation sweep. Thus, by examining these coefficients, we can determine how quickly the error tends to zero, which indicates how quickly the solution converges. The following analysis does not apply to points on the boundary, it is only valid for interior points. Thus, while it gives more information about the functioning of the scheme in the interior of the domain, boundary peculiarities are not addressed.

We begin by recalling that the current **error**, $\vec{e}_{(s)}$, is the difference between the exact solution, $\vec{T}^*$, and the current approximation, $\vec{T}_{(s)}$,

$$\vec{e}_{(s)} = \vec{T}^* - \vec{T}_{(s)},$$

where we desire that the sequence, $\{\vec{e}_{(s)}\}$, tend to zero. Substituting this expression into Equations IV.6 and IV.11, we have the following error equations for the Jacobi and Gauss-Seidel methods where, in order to avoid confusion with the exponential function in the DFT expansion, we represent the components of $\vec{e}$ by $v_{k,j}$:

$$\text{Jacobi} \quad : \quad v_{k,j}^{(new)} = -\frac{\frac{h^3}{g384}(\hat{\sum}_V^{old}) - \frac{\alpha\kappa h}{2}(\hat{\sum}_S^{old})}{\frac{h^3}{g384}224j + \frac{\alpha\kappa h}{2}8j} \tag{IV.22}$$

$$\text{Gauss} - \text{Seidel} \quad : \quad v_{k,j}^{(new)} = -\frac{\frac{h^3}{g384}(\check{\sum}_V) - \frac{\alpha\kappa h}{2}(\check{\sum}_S)}{\frac{h^3}{g384}224j + \frac{\alpha\kappa h}{2}8j}. \tag{IV.23}$$

where $\hat{\sum}_V$, $\hat{\sum}_S$, $\check{\sum}_V$, and $\check{\sum}_S$ are now applied to the $v_{k,j}$ (see Equations IV.7, IV.8, IV.12, and IV.13). Equations IV.22 and IV.23 indicate how the sequence $\{\vec{e}_{(s)}\}$ evolves during the iteration process. If we expand these relationships in a discrete Fourier transform, (DFT) (see Appendix A or [Ref. 10]), the magnitude of the transform coefficients will indicate the "amount" of each mode of the error that is present in each component of $\vec{e}_{(s)}$. We then compare coefficients to determine the ratio between the new and current approximations for each component of the error. In other words, the DFT allows us to analyze the growth of the error by examining component behavior.

Expanding Equation IV.22 in a DFT, where

$$v_{k,j} = \sum\sum_{l,m=-\frac{N}{2}+1}^{\frac{N}{2}} V_{l,m} e^{\frac{\imath 2\pi kl}{N}} e^{\frac{\imath 2\pi jm}{N}},$$

$$C(l,m) = e^{\frac{\imath 2\pi kl}{N}} e^{\frac{\imath 2\pi jm}{N}} \neq 0, \quad \text{and} \quad \mathcal{F}(j) = \frac{1}{\frac{h^3}{g384}224j + \frac{\alpha\kappa h}{2}8j},$$

and the superscripts (o) and (n) indicate the components of the old and new approximations, we have

$$\begin{aligned}
\sum\sum_{l,m=-\frac{N}{2}+1}^{\frac{N}{2}} V_{l,m}^{(n)} C(l,m) \;=\; & -\Bigg\{\sum\sum_{l,m=-\frac{N}{2}+1}^{\frac{N}{2}} \Bigg[\frac{h^3}{g384}((32j-5)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi l}{N}} \\
& +(16j+5)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi(l+m)}{N}} \\
& +(32j-11)V_{l,m}^{(o)}C(l,m)e^{\frac{-\imath 2\pi m}{N}} \\
& +(32j+11)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi m}{N}} \\
& +(16j-5)V_{l,m}^{(o)}C(l,m)e^{-\frac{\imath 2\pi(l+m)}{N}} \\
& +(32j+5)V_{l,m}^{(o)}C(l,m)e^{-\frac{\imath 2\pi l}{N}}) \\
& -\frac{\alpha\kappa h}{2}(2jV_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi l}{N}} \\
& +(-1+2j)V_{l,m}^{(o)}C(l,m)e^{-\frac{\imath 2\pi m}{N}} \\
& +(1+2j)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi m}{N}} \\
& +2jV_{l,m}^{(o)}C(l,m)e^{\frac{-\imath 2\pi l}{N}})\Bigg]\Bigg\}\mathcal{F}(j),
\end{aligned}$$

Making use of the orthogonality property of the complex exponential (see Appendix A) we can equate individual terms of the sum, and then divide by $C(l,m)$ to give

$$\begin{aligned}
V_{l,m}^{(n)} \;=\; & -\Bigg[\frac{h^3}{g384}((32j-5)V_{l,m}^{(o)}e^{\frac{\imath 2\pi l}{N}} + (16j+5)V_{l,m}^{(o)}e^{\frac{\imath 2\pi(l+m)}{N}} \\
& +(32j-11)V_{l,m}^{(o)}e^{\frac{-\imath 2\pi m}{N}} + (32j+11)V_{l,m}^{(o)}e^{\frac{\imath 2\pi m}{N}} \\
& +(16j-5)V_{l,m}^{(o)}e^{-\frac{\imath 2\pi(l+m)}{N}} + (32j+5)V_{l,m}^{(o)}e^{-\frac{\imath 2\pi l}{N}}) \\
& -\frac{\alpha\kappa h}{2}(2jV_{l,m}^{(o)}e^{\frac{\imath 2\pi l}{N}} + (-1+2j)V_{l,m}^{(o)}e^{-\frac{\imath 2\pi m}{N}} \\
& +(1+2j)V_{l,m}^{(o)}e^{\frac{\imath 2\pi m}{N}} + 2jV_{l,m}^{(o)}e^{\frac{-\imath 2\pi l}{N}})\Bigg]\mathcal{F}(j),
\end{aligned}$$

or

$$\begin{aligned}
V_{l,m}^{(n)} \;=\; & -\Bigg[\frac{h^3}{g384}((32j-5)e^{\frac{\imath 2\pi l}{N}} + (16j+5)e^{\frac{\imath 2\pi(l+m)}{N}} \\
& +(32j-11)e^{\frac{-\imath 2\pi m}{N}} + (32j+11)e^{\frac{\imath 2\pi m}{N}} \\
& +(16j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} + (32j+5)e^{-\frac{\imath 2\pi l}{N}}) \\
& -\frac{\alpha\kappa h}{2}(2je^{\frac{\imath 2\pi l}{N}} + (-1+2j)e^{-\frac{\imath 2\pi m}{N}} \\
& +(1+2j)e^{\frac{\imath 2\pi m}{N}} + 2je^{\frac{-\imath 2\pi l}{N}})\Bigg]\mathcal{F}(j)V_{l,m}^{(o)}.
\end{aligned}$$

	N = 16			N = 32		
	$j=1$	$j=\frac{N}{2}$	$j=N-1$	$j=1$	$j=\frac{N}{2}$	$j=N-1$
$\alpha=\frac{1}{2}$	1.009	1.009	1.009	1.026	1.026	1.026
$\alpha=1$	1.005	1.005	1.005	1.013	1.013	1.013

Table III. Maximum Values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $l,m=0:N$ for the Jacobi Method of Relaxation.

Therefore, the ratio of the new coefficient to the old is

$$\begin{aligned}\frac{V_{l,m}^{(n)}}{V_{l,m}^{(o)}} &= -\Big[\frac{h^3}{g384}((32j-5)e^{\frac{\imath 2\pi l}{N}}+(16j+5)e^{\frac{\imath 2\pi(l+m)}{N}} \\ &\quad +(32j-11)e^{\frac{-\imath 2\pi m}{N}}+(32j+11)e^{\frac{\imath 2\pi m}{N}} \\ &\quad +(16j-5)e^{-\frac{\imath 2\pi(l+m)}{N}}+(32j+5)e^{-\frac{\imath 2\pi l}{N}}) \\ &\quad -\frac{\alpha\kappa h}{2}(2je^{\frac{\imath 2\pi l}{N}}+(-1+2j)e^{-\frac{\imath 2\pi m}{N}} \\ &\quad +(1+2j)e^{\frac{\imath 2\pi m}{N}}+2je^{\frac{-\imath 2\pi l}{N}})\Big]\mathcal{F}(j).\end{aligned} \tag{IV.24}$$

In order to determine the greatest factor by which a mode of the error is multiplied, we seek the maximum of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ over the values $l,m=-\frac{N}{2}+1:\frac{N}{2}$. In other words, we seek to determine a bound on how well we can expect this type of relaxation scheme to perform. This ratio is a function of N,l,m and j and, while difficult to determine analytically, may be calculated numerically. Using *Matlab*, the maximum of this ratio for $l,m=0:N$ has been calculated for $N=16,32$; $\alpha=\frac{1}{2},1$; and $j=1,\frac{N}{2},N-1$; the results are indicated in Table III (see Appendix A for a discussion of the equivalence of centered indices, $l,m=-\frac{N}{2}+1:\frac{N}{2}$, and non-centered indices, $l,m=0:N$). Additionally, by considering the matrix of grid points $l,m=0:N$, we can determine a correspondence between sample points on the grid and type of associated frequency (see Figure 11 and Appendix A). The matrices of values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $j=N-1$ are depicted in Figures 12 and 13.

The information in Table III indicates that the Jacobi method results in a

maximum of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ greater than one; this value does not appear to depend on grid position. In other words, there is at least some component of the error that is magnified, instead of reduced, by this iteration scheme. Moreover, it appears that the value increases with the number of intervals on the grid. Additionally, Figures 12 and 13 indicate that the maximum of this ratio occurs for both low and high frequencies. Thus, it appears that the Jacobi method will not be effective in generating a solution to the point-source problem.

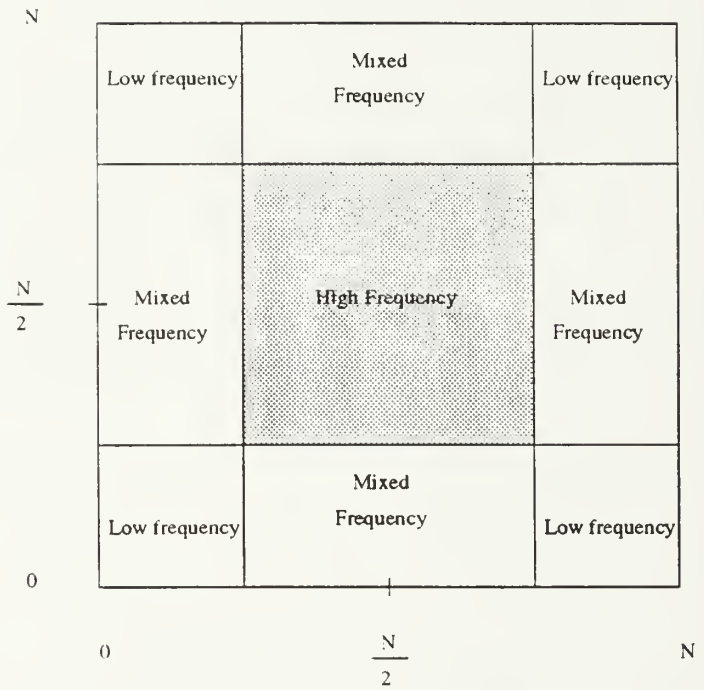

Figure 11. A two-dimensional representation of frequencies corresponding to a single set of non-centered sample points.

In order to compute the ratio $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for the Gauss-Seidel relaxation scheme, we rewrite Equation IV.23 as

$$\begin{aligned} \frac{h^3}{g384} D v_{k,j}^{(new)} - \frac{\alpha\kappa h}{2} J v_{k,j}^{(new)} \;=\; & \frac{h^3}{g384}(-A v_{k-1,j-1}^{(new)} - B v_{k-1,j}^{(new)} - C v_{k,j-1}^{(new)} \\ & -E v_{k,j+1}^{(old)} - F v_{k+1,j}^{(old)} - G v_{k+1,j+1}^{(old)}) \\ & -\frac{\alpha\kappa h}{2}(-H v_{k-1,j}^{(new)} - I v_{k,j-1}^{(new)} - K v_{k,j+1}^{(old)} - L v_{k+1,j}^{(old)}), \end{aligned} \qquad \text{(IV.25)}$$

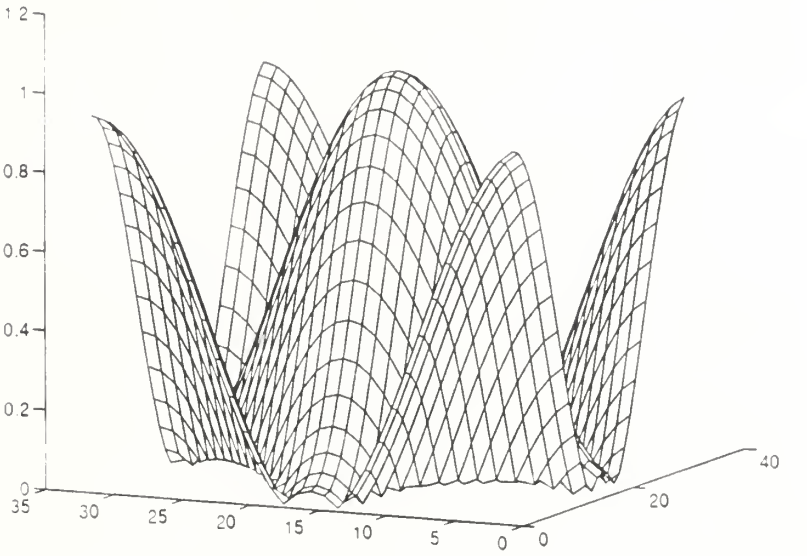

Figure 12. Ratio of amplitudes of new to old Fourier coefficients, Jacobi relaxation: $N = 32$, $\alpha = \frac{1}{2}$, $j = N - 1$. Frequency ranges are as in Figure 11.

where $A = 16j-5, B = 32j+5, C = 32j-11, D = 224j, E = 32j+11, F = 32j-5, G = 16j + 5$, and $H = 2j, I = -1 + 2j, J = -8j, K = 1 + 2j, L = 2j$. Expanding in a DFT, equating terms in the (double) sum by virtue of the orthogonality property of the complex exponential (see Appendix A), and dividing by $C(l, m)$, we have

$$\begin{aligned}\frac{h^3}{g384} DV_{l,m}^{(n)} - \frac{\alpha\kappa h}{2} JV_{l,m}^{(n)} \;=\; & \frac{h^3}{g384}\left(-AV_{l,m}^{(n)} e^{-\frac{\imath 2\pi(l+m)}{N}} - BV_{l,m}^{(n)} e^{-\frac{\imath 2\pi l}{N}} - CV_{l,m}^{(n)} e^{\frac{-\imath 2\pi m}{N}}\right. \\ & \left. -EV_{l,m}^{(o)} e^{\frac{\imath 2\pi m}{N}} - FV_{l,m}^{(o)} e^{\frac{\imath 2\pi l}{N}} - GV_{l,m}^{(o)} e^{\frac{\imath 2\pi(l+m)}{N}}\right) \\ & -\frac{\alpha\kappa h}{2}\left(-HV_{l,m}^{(n)} e^{\frac{-\imath 2\pi l}{N}} - IV_{l,m}^{(n)} e^{-\frac{\imath 2\pi m}{N}}\right. \\ & \left. -KV_{l,m}^{(o)} e^{\frac{\imath 2\pi m}{N}} - LV_{l,m}^{(o)} e^{\frac{\imath 2\pi l}{N}}\right),\end{aligned}$$

or, regrouping we have

$$\begin{aligned}\left\{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}} + Be^{-\frac{\imath 2\pi l}{N}} + Ce^{\frac{-\imath 2\pi m}{N}} + D]\right. \\ \left. -\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}} + Ie^{-\frac{\imath 2\pi m}{N}} + J]\right\} V_{l,m}^{(n)}\end{aligned}$$

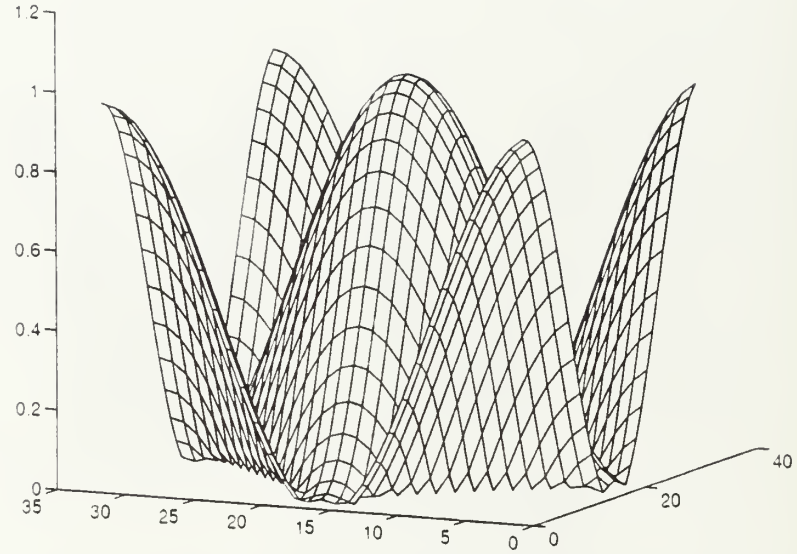

Figure 13. Ratio of amplitudes of new to old Fourier coefficients, Jacobi relaxation: $N = 32$, $\alpha = 1$, $j = N - 1$. Frequency ranges are as in Figure 11.

$$= \left\{ \frac{h^3}{g384}[-Ee^{\frac{\imath 2\pi m}{N}} - Fe^{\frac{\imath 2\pi l}{N}} - Ge^{\frac{\imath 2\pi(l+m)}{N}}] - \frac{\alpha\kappa h}{2}[-Ke^{\frac{\imath 2\pi m}{N}} - Le^{\frac{\imath 2\pi l}{N}}] \right\} V_{l,m}^{(o)}.$$

Thus, the ratio $\frac{V_{l,m}^{(n)}}{V_{l,m}^{(o)}}$ is given by

$$\frac{\frac{h^3}{g384}[-Ee^{\frac{\imath 2\pi m}{N}} - Fe^{\frac{\imath 2\pi l}{N}} - Ge^{\frac{\imath 2\pi(l+m)}{N}}] - \frac{\alpha\kappa h}{2}[-Ke^{\frac{\imath 2\pi m}{N}} - Le^{\frac{\imath 2\pi l}{N}}]}{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}} + Be^{-\frac{\imath 2\pi l}{N}} + Ce^{\frac{-\imath 2\pi m}{N}} + D] - \frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}} + Ie^{-\frac{\imath 2\pi m}{N}} + J]}. \quad \text{(IV.26)}$$

As above, we now seek to maximize $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ over the values $l, m = -\frac{N}{2} + 1 : \frac{N}{2}$ in order to determine a bound on how well this scheme performs. We again use *Matlab* to compute the maximum of this ratio for $l, m = 0 : N$ for $N = 16, 32$; $\alpha = \frac{1}{2}, 1$; and $j = 1, \frac{N}{2}, N - 1$; the results are indicated in Table IV. Additionally, the matrices of values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $j = N - 1$ are depicted in Figures 14 and 15.

The information in Table IV indicates that the Gauss-Seidel method results in a maximum of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ less than one. Thus, all frequency components of the error are reduced by this relaxation scheme, which is what we seek. However, the value

	N = 16			N = 32		
	$j=1$	$j=\frac{N}{2}$	$j=N-1$	$j=1$	$j=\frac{N}{2}$	$j=N-1$
$\alpha=\frac{1}{2}$	.8986	.8796	.8781	.9152	.8977	.8970
$\alpha=1$	.9478	.9374	.9366	.9567	.9473	.9469

Table IV. Maximum Values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $l,m=0:N$ for the Gauss-Seidel Method of Relaxation.

increases with grid size N, and also as the time stepping is shifted from Crank-Nicholson to fully implicit. Additionally, while all of the values are less than one, as we move to grids with larger N and toward a fully implicit time scheme, they become close to unity. This means that, although we may reasonably expect to see this relaxation process converge, it may be quite slow. Another point is that the maximum value depends on grid position; the shorter the radial distance, the larger the maximum, which apparently is a reflection of the radial bias of the stencils. Additionally, Figures 14 and 15 indicate that the maximum of this ratio occurs only over the low frequencies, and that the values associated with error components over the high frequencies are quite low. This performance over the high frequencies will be of importance in Chapter VI. Thus, it appears that the Gauss-Seidel method will be effective in generating an iterative solution to the point-source problem, albeit potentially slow to converge.

We now apply similar techniques to the radial/vertical line relaxation schemes. The error equations come from rewriting Equations IV.20 and IV.21 respectively as

$$\begin{aligned}\frac{h^3}{g384}Dv_{k,j}^{(new)}-\frac{\alpha\kappa h}{2}Jv_{k,j}^{(new)} &= \frac{h^3}{g384}\left(-Av_{k-1,j-1}^{(new)}-Bv_{k-1,j}^{(new)}-Cv_{k,j-1}^{(new)}\right.\\ &\quad \left.-Ev_{k,j+1}^{(new)}-Fv_{k+1,j}^{(old)}-Gv_{k+1,j+1}^{(old)}\right)\\ &\quad -\frac{\alpha\kappa h}{2}\left(-Hv_{k-1,j}^{(new)}-Iv_{k,j-1}^{(new)}-Kv_{k,j+1}^{(new)}-Lv_{k+1,j}^{(old)}\right),\end{aligned}$$

and

$$\begin{aligned}\frac{h^3}{g384}Dv_{k,j}^{(new)}-\frac{\alpha\kappa h}{2}Jv_{k,j}^{(new)} &= \frac{h^3}{g384}\left(-Av_{k-1,j-1}^{(new)}-Bv_{k-1,j}^{(new)}-Cv_{k,j-1}^{(new)}\right.\end{aligned}$$

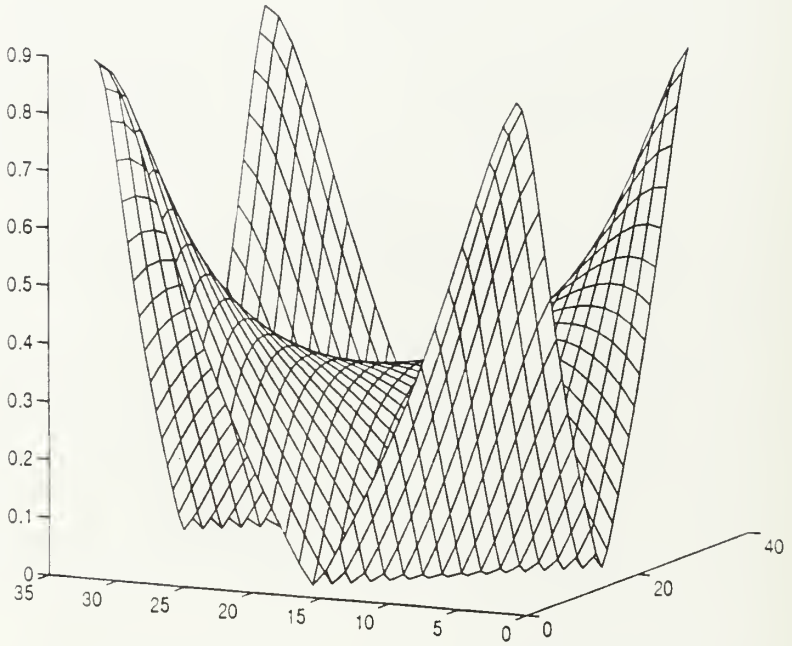

Figure 14. Ratio of amplitudes of new to old Fourier coefficients, Gauss-Seidel relaxation: $N = 32$, $\alpha = \frac{1}{2}$, $j = N - 1$. Frequency ranges are as in Figure 11.

$$
\begin{aligned}
&- Ev_{k,j+1}^{(old)} - Fv_{k+1,j}^{(new)} - Gv_{k+1,j+1}^{(old)}\Big) \\
&-\frac{\alpha\kappa h}{2}\left(-Hv_{k-1,j}^{(new)} - Iv_{k,j-1}^{(new)} - Kv_{k,j+1}^{(old)} - Lv_{k+1,j}^{(new)}\right).
\end{aligned}
$$

As above, we expand these relations in a DFT, make use of the orthogonality property to equate terms in the sums, and divide by $C(l,m)$ to obtain

$$
\begin{aligned}
\frac{h^3}{g384}DV_{l,m}^{(n)} - \frac{\alpha\kappa h}{2}JV_{l,m}^{(n)} \;=\; & \frac{h^3}{g384}\left(-AV_{l,m}^{(n)}e^{-\frac{i2\pi(l+m)}{N}} - BV_{l,m}^{(n)}e^{-\frac{i2\pi l}{N}} - CV_{l,m}^{(n)}e^{\frac{-i2\pi m}{N}}\right. \\
& \left. - EV_{l,m}^{(n)}e^{\frac{i2\pi m}{N}} - FV_{l,m}^{(o)}e^{\frac{i2\pi l}{N}} - GV_{l,m}^{(o)}e^{\frac{i2\pi(l+m)}{N}}\right) \\
& -\frac{\alpha\kappa h}{2}\left(-HV_{l,m}^{(n)}e^{\frac{-i2\pi l}{N}} - IV_{l,m}^{(n)}e^{-\frac{i2\pi m}{N}}\right. \\
& \left. - KV_{l,m}^{(n)}e^{\frac{i2\pi m}{N}} - LV_{l,m}^{(o)}e^{\frac{i2\pi l}{N}}\right),
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{h^3}{g384}DV_{l,m}^{(n)} - \frac{\alpha\kappa h}{2}JV_{l,m}^{(n)} \;=\; & \frac{h^3}{g384}\left(-AV_{l,m}^{(n)}e^{-\frac{i2\pi(l+m)}{N}} - BV_{l,m}^{(n)}e^{-\frac{i2\pi l}{N}} - CV_{l,m}^{(n)}e^{\frac{-i2\pi m}{N}}\right. \\
& \left. - EV_{l,m}^{(o)}e^{\frac{i2\pi m}{N}} - FV_{l,m}^{(n)}e^{\frac{i2\pi l}{N}} - GV_{l,m}^{(o)}e^{\frac{i2\pi(l+m)}{N}}\right)
\end{aligned}
$$

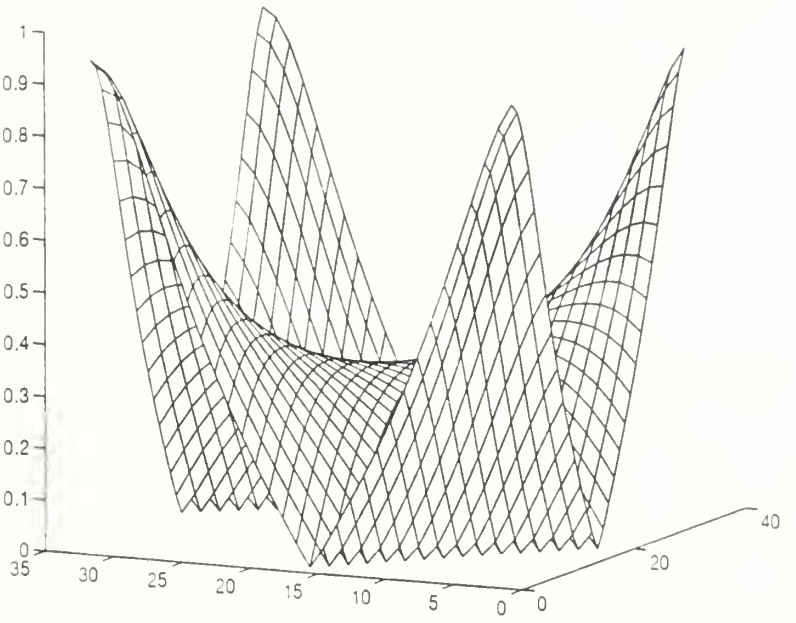

Figure 15. Ratio of amplitudes of new to old Fourier coefficients, Gauss-Seidel relaxation: $N = 32$, $\alpha = 1$, $j = N - 1$. Frequency ranges are as in Figure 11.

$$-\frac{\alpha\kappa h}{2}\left(-HV_{l,m}^{(n)}e^{\frac{-\imath 2\pi l}{N}} - IV_{l,m}^{(n)}e^{-\frac{\imath 2\pi m}{N}}\right.$$
$$\left. - KV_{l,m}^{(o)}e^{\frac{\imath 2\pi m}{N}} - LV_{l,m}^{(n)}e^{\frac{\imath 2\pi l}{N}}\right).$$

Regrouping, we have

$$\{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}} + Be^{-\frac{\imath 2\pi l}{N}} + Ce^{\frac{-\imath 2\pi m}{N}} + D + Ee^{\frac{\imath 2\pi m}{N}}]$$
$$-\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}} + Ie^{-\frac{\imath 2\pi m}{N}} + J + Ke^{\frac{\imath 2\pi m}{N}}]\}V_{l,m}^{(n)}$$
$$= \{\frac{h^3}{g384}[-Fe^{\frac{\imath 2\pi l}{N}} - Ge^{\frac{\imath 2\pi(l+m)}{N}}] - \frac{\alpha\kappa h}{2}[-Le^{\frac{\imath 2\pi l}{N}}]\}V_{l,m}^{(o)},$$

and

$$\{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}} + Be^{-\frac{\imath 2\pi l}{N}} + Ce^{\frac{-\imath 2\pi m}{N}} + D + Fe^{\frac{\imath 2\pi l}{N}}]$$
$$-\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}} + Ie^{-\frac{\imath 2\pi m}{N}} + J + Le^{\frac{\imath 2\pi l}{N}}]\}V_{l,m}^{(n)}$$
$$= \{\frac{h^3}{g384}[-Ee^{\frac{\imath 2\pi m}{N}} - Ge^{\frac{\imath 2\pi(l+m)}{N}}] - \frac{\alpha\kappa h}{2}[-Ke^{\frac{\imath 2\pi m}{N}}]\}V_{l,m}^{(o)}.$$

	N = 16			N = 32		
	$j=1$	$j=\frac{N}{2}$	$j=N-1$	$j=1$	$j=\frac{N}{2}$	$j=N-1$
$\alpha=\frac{1}{2}$	.7689	.7723	.7726	.8011	.8051	.8052
$\alpha=1$	.8758	.8768	.8768	.8954	.8965	.8965

Table V. Maximum Values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $l,m=0:N$ for Radial Line Relaxation.

Thus, the numerators of the ratio $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ are given by

$$\frac{h^3}{g384}[-Fe^{\frac{\imath 2\pi l}{N}}-Ge^{\frac{\imath 2\pi(l+m)}{N}}]-\frac{\alpha\kappa h}{2}[-Le^{\frac{\imath 2\pi l}{N}}]$$

and

$$\frac{h^3}{g384}[-Ee^{\frac{\imath 2\pi m}{N}}-Ge^{\frac{\imath 2\pi(l+m)}{N}}]-\frac{\alpha\kappa h}{2}[-Ke^{\frac{\imath 2\pi m}{N}}],$$

and the denominators are given by

$$\{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}}+Be^{-\frac{\imath 2\pi l}{N}}+Ce^{\frac{-\imath 2\pi m}{N}}+D+Ee^{\frac{\imath 2\pi m}{N}}]$$
$$-\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}}+Ie^{-\frac{\imath 2\pi m}{N}}+J+Ke^{\frac{\imath 2\pi m}{N}}]$$

and

$$\{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}}+Be^{-\frac{\imath 2\pi l}{N}}+Ce^{\frac{-\imath 2\pi m}{N}}+D+Fe^{\frac{\imath 2\pi l}{N}}]$$
$$-\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}}+Ie^{-\frac{\imath 2\pi m}{N}}+J+Le^{\frac{\imath 2\pi l}{N}}].$$

As above, we now seek to maximize $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ over the values $l,m=-\frac{N}{2}+1:\frac{N}{2}$. We continue to use *Matlab* to compute the maximum of these ratios for $l,m=0:N$ for $N=16,32$; $\alpha=0,\frac{1}{2},1$; and $j=1,\frac{N}{2},N-1$; the results are indicated in Tables V and VI. Additionally, the matrices of values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $j=N-1$ are depicted in Figures 16, 17, 18, and 19.

The information in Tables V and VI indicates that both line relaxation methods result in a maximum of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ less than one; radial line relaxation seems to promise

	N = 16			N = 32		
	$j=1$	$j=\frac{N}{2}$	$j=N-1$	$j=1$	$j=\frac{N}{2}$	$j=N-1$
$\alpha=\frac{1}{2}$	.8379	.7837	.7787	.8627	.8103	.8079
$\alpha=1$	.9149	.8834	.8805	.9290	.8994	.8981

Table VI. Maximum Values of $\frac{|V_{l,m}^{(n)}|}{|V_{l,m}^{(o)}|}$ for $l,m=0:N$ for Vertical Line Relaxation.

better performance. Many of the results for the line relaxation schemes parallel those noted for the Gauss-Seidel method. All frequency components of the error are reduced by the line relaxation schemes; their performance would appear to be slightly better than that for the Gauss-Seidel method. Additionally, the maximum values increase with both the grid size N, and as the time stepping is shifted from Crank-Nicholson to fully implicit. As with the Gauss-Seidel method, while all of the values are less than one, as we move to grids with larger N and toward a fully implicit time scheme, they become close to unity. This means that, although we may reasonably expect to see this relaxation process converge, it may be quite slow, and that the increased amount of computational work as compared with work for the Gauss-Seidel method may not be worth the payoff. The maximum values for these schemes also depend on grid position. For the horizontal line relaxation, the shorter the radial distance, the smaller the maximum; for the vertical line relaxation, the shorter the radial distance, the larger the maximum. Once again, apparently, this is a reflection of the radial bias of the stencils. Additionally, Figures 16, 17, 18, and 19 indicate that the maximum of these ratios occurs only over low frequencies, and that the values associated with error components over the high frequencies are quite low. In fact, the values for the line relaxation schemes over the high frequencies appear to be about one-half the corresponding values for the Gauss-Seidel method. As noted earlier, this performance over the high frequencies will be of importance in Chapter VI.

Some general trends are evident from the results of the numerical experiments.

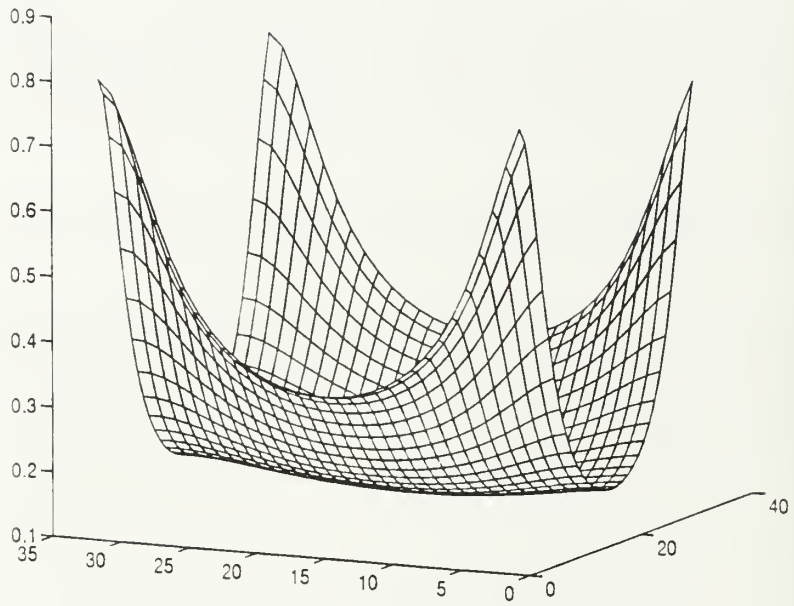

Figure 16. Ratio of amplitudes of new to old Fourier coefficients, radial line relaxation: $N = 32$, $\alpha = \frac{1}{2}$, $j = N - 1$. Frequency ranges are as in Figure 11.

Since the maxima should all be less than one to guarantee convergence, it appears that the Jacobi method will not be useful in iteratively solving the point-source problem (Equation II.2). Of the schemes that have maxima less than one, those that have the smallest maxima are those that reduce error components most quickly and, therefore, are the schemes that will converge most quickly. The line relaxation schemes have the smallest maxima; the maximum values over the high frequencies are about one-half the corresponding values for the Gauss-Seidel method, but there is little increase in performance over the low frequencies. To determine whether or the extra work in using line relaxation is worth the effort, a cost comparison of getting to the same level of error as other relaxation schemes will need to be done. Additionally, the maximum values vary depending on the time stepping scheme and the grid position; they are higher for implicit time-stepping than for Crank-Nicholson, indicating that convergence will take longer for the former as compared to the latter, and the maximum values increase with N. Hence, as we move to a grid with more nodes, we not

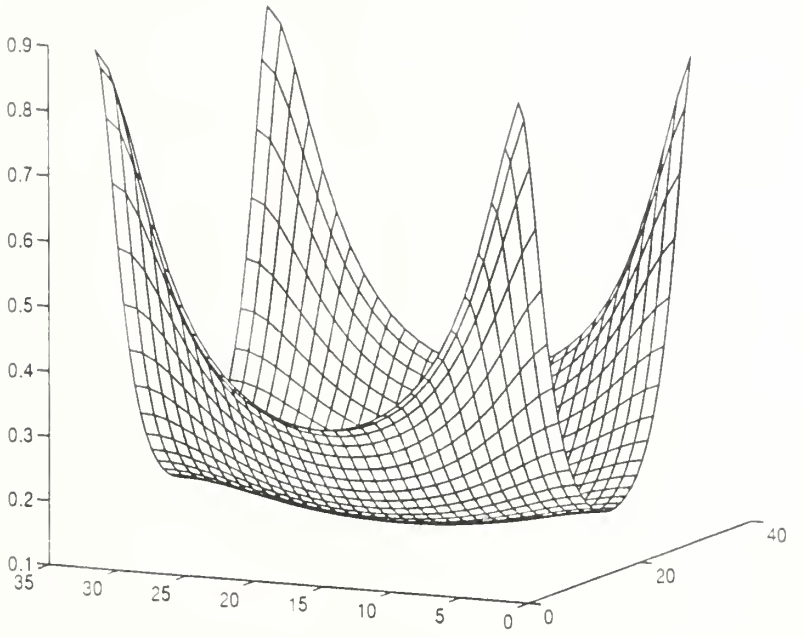

Figure 17. Ratio of amplitudes of new to old Fourier coefficients, radial line relaxation: $N = 32$, $\alpha = 1$, $j = N - 1$. Frequency ranges are as in Figure 11.

only have more work to do by virtue of the greater number of equations to solve, we also have to work harder to reduce each component of the error. That is, the effect of moving to a finer grid is to more than quadruple the work required to solve the problem.

The Gauss-Seidel and line relaxation schemes appear to be the most efficient in terms of reducing error components. Additionally, in Chapter VI, the performance of these schemes over the high frequencies will be of interest; we will want to use those schemes that have the smallest maximum value over the high frequencies. In particular, for the Gauss-Seidel and line relaxation schemes, the high frequency components of the error are eliminated much more quickly than the low frequency components. Considering the amount of work to be done and the performance of the scheme, we implement the Gauss-Seidel method in the solution process.

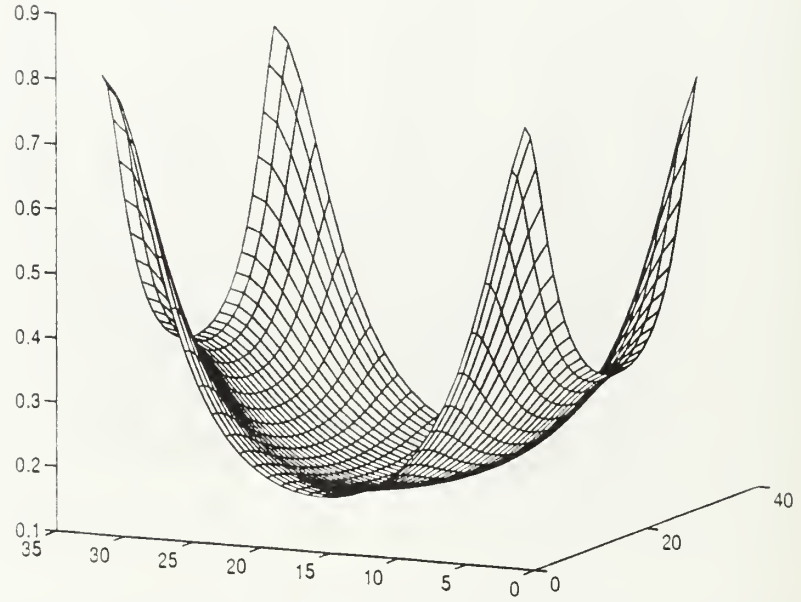

Figure 18. Ratio of amplitudes of new to old Fourier coefficients, vertical line relaxation: $N = 32$, $\alpha = \frac{1}{2}$, $j = N - 1$. Frequency ranges are as in Figure 11.

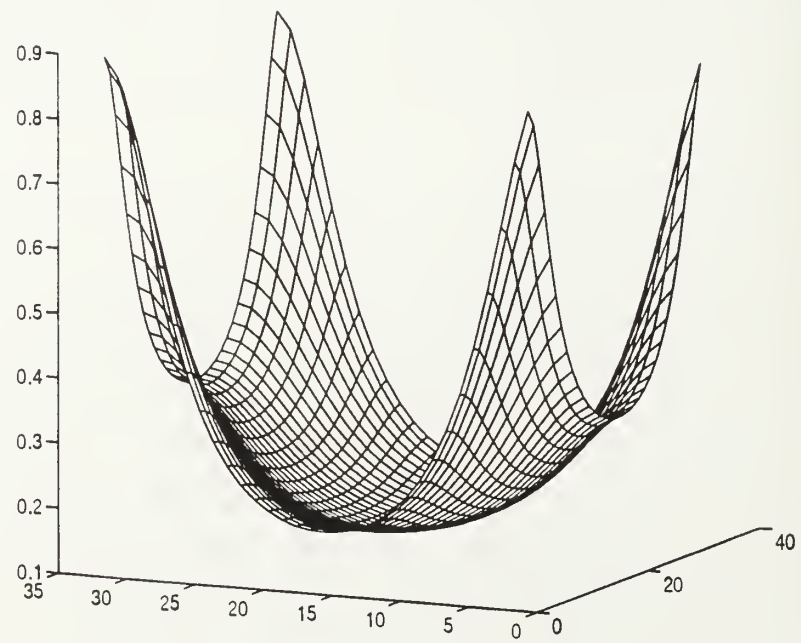

Figure 19. Ratio of amplitudes of new to old Fourier coefficients, vertical line relaxation: $N = 32$, $\alpha = 1$, $j = N - 1$. Frequency ranges are as in Figure 11.

V. ITERATIVE SOLUTION

The results of Chapter IV indicate that the Gauss-Seidel method is the simplest of the relaxation schemes that appear to result in a converging sequence of approximate solutions, and therefore Gauss-Seidel is our method of choice. Having chosen a relaxation scheme, we must now, as part of the iterative solution process, make suitable choices for the type of time-stepping, α, and the time step size, g; the goal is to maximize accuracy and minimize computational work. We now conduct experiments to solve Equation IV.3 by iterating with a solver which is based on the iteration matrix, $\mathbf{P}_G = -(\mathbf{D} + \mathbf{L})^{-1}\mathbf{U}$, and use this process to evaluate some of the possible choices for α and g. The technique is to solve first at a single time step by iterating until the difference between successive approximations, $\|\vec{T}^n_{(s+1)} - \vec{T}^n_{(s)}\|$ or, perhaps better, $\|\mathbf{M}\vec{T}^n_{(s+1)} - \vec{f}(\vec{T}^n_{(s)})\|$, is negligible. This approximate solution, $\vec{T}^n$, is used as input for the time-stepping scheme so that

$$\vec{f}(\vec{T}^n) = \sum_{l=1}^{(N+1)^2} [\frac{1}{g}\int_V \phi^n_l dV + (1-\alpha)\kappa \int_S \nabla\phi^n_l \cdot \hat{n} dS]T^n_l$$

(as in Chapter III) becomes the new right hand side in Equation IV.3. The iteration process is repeated to obtain the approximation at the new time step. The solution is stepped in time in this fashion until the difference between solutions at successive time steps is negligible, representing a steady solution.

An algorithm for this process might look something like the following:

Given an initial guess, $\vec{T}^0$, and tolerances δ_1, δ_2,
Set $n = 0$
While $\|\vec{T}^{n+1} - \vec{T}^n\| > \delta_2$
1) Compute $\vec{f}(\vec{T}^n)$
2) Set $s = 1$, $\vec{T}^{n+1}_{(0)} = \vec{T}^n$
3) While $\|\vec{T}^{n+1}_{(s)} - \vec{T}^{n+1}_{(s-1)}\| > \delta_1$
$\vec{T}^{n+1}_{(s+1)} \leftarrow \mathbf{P}_G \vec{T}^{n+1}_{(s)} + (\mathbf{D} + \mathbf{L})^{-1} \vec{f}(\vec{T}^n)$
$s = s + 1$
End while
4) $\vec{T}^{n+1} \leftarrow \vec{T}^{n+1}_{(s)}$
Return $\vec{T}^{n+1}$ as the solution at $t = (n+1)\Delta t$
$n = n + 1$
End while
$\vec{T}^{steady} \leftarrow \vec{T}^n$
Stop

Starting with an initial temperature distribution of $T = 0$ everywhere, boundary conditions specified by Equations II.18, II.19, and II.20, $N = 16$, and $\alpha = \frac{1}{2}$, several values of g are used in an attempt to achieve a reasonable solution. The results of these experiments indicate that for larger values of g, say $g = h$, the solution exhibits a non-physical oscillation in time (see Figure 20) at the origin, where the heat source is located. Instead, the solution should monotonically increase until it levels off at steady state.

The oscillation can be removed by making the temporal step size smaller, say $g = h^3$ (see Figure 21). This step size, however, results in negative temperatures along the $z = 0$ axis near the origin. Negative temperatures, like the oscillations, are physically meaningless. A valid solution process must eliminate both of these non-physical characteristics from the solution. However, it turns out that for $\alpha = \frac{1}{2}$, there is no value of g that produces a physically meaningful solution. In particular, for $g = h^{\frac{5}{2}}$, there are both oscillations and negative values in the approximate solution (see Figure 22 for a comparison of several values of g). Thus, in order to compute a realistic solution, it is necessary to use values of $\alpha > \frac{1}{2}$.

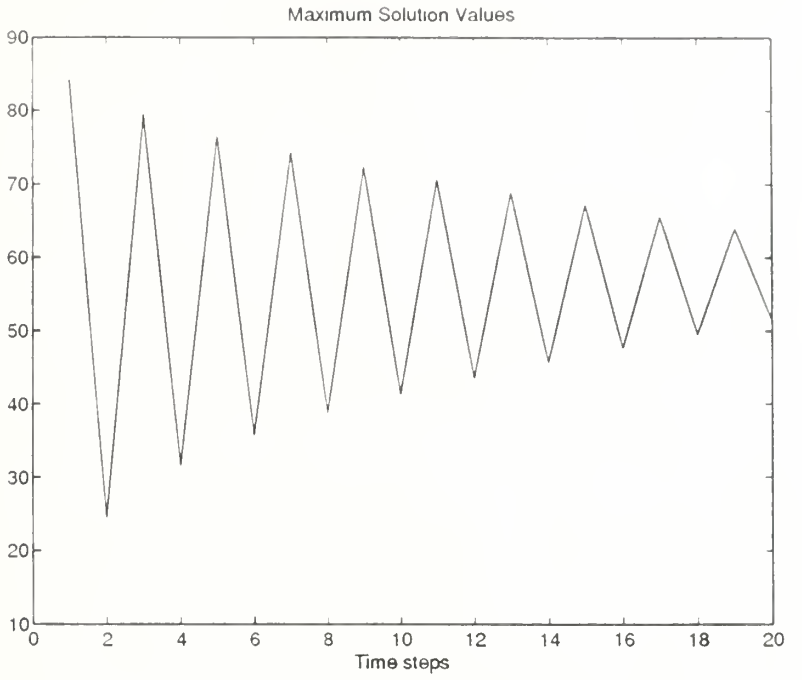

Figure 20. The maximum values of the solution for each time step, where $\alpha = \frac{1}{2}$ and $g = h$.

In an attempt to produce a solution that is physically meaningful, numerical experiments are conducted using values of $\frac{1}{2} \leq \alpha \leq 1$, and values of g ranging from h^2 to h for each value of α. Solutions with no oscillation and no negative values are possible for, say $\alpha = \frac{3}{4}$, but this requires that the temporal step size be as small as $g = h^{\frac{1111}{512}}$. The size of the temporal step indicates how much work will have to be done to reach a steady solution: the smaller the step, the longer to reach steady state. In order to minimize the work, we must maximize the temporal step size. Further experimentation indicates that for $\alpha = 1$ (fully implicit time stepping) and $g = h$, the result is not only a physically meaningful solution (i.e., no oscillations and no negative values), but also a reasonable temporal step size. Therefore, the iterative solution is computed using these values.

Iterative solutions for $N = 8, 16$, and 32, are computed using the same initial temperature distribution and boundary conditions as above. One measure of accuracy of these solutions is determined by comparing them with an analytic solution

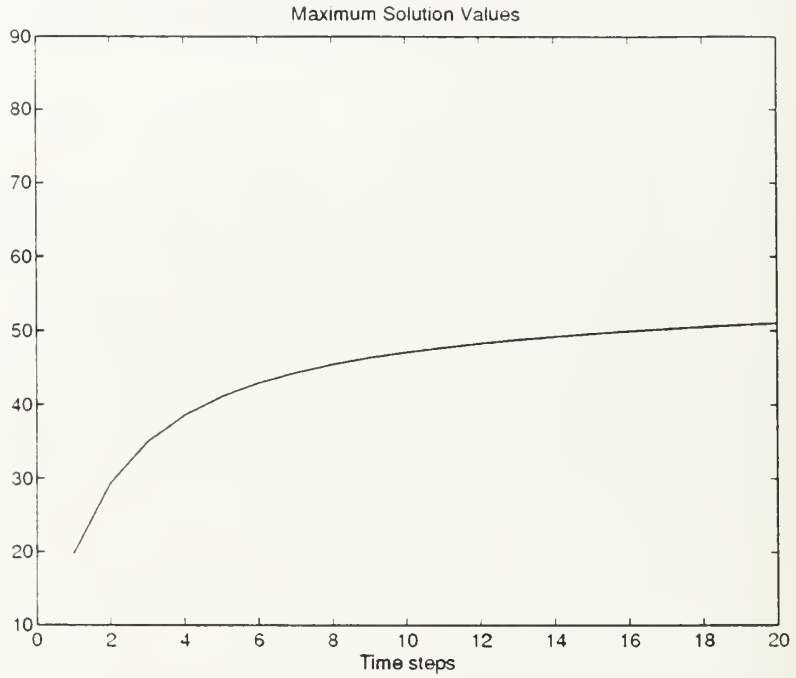

Figure 21. The maximum values of the solution for each time step, where $\alpha = \frac{1}{2}$ and $g = h^3$.

(Equations II.15 and II.16). However, the problem that gives rise to the analytic solution presented in Chapter II is somewhat different from the linear algebra problem that we are solving. Therefore, as in [Ref. 11] we approximate the exact solution of Equation II.2 by solving Equation IV.3 for $N = 64$ (see Figure 23). We then compare this solution with solutions of Equation IV.3 on coarser grids ($N = 32$ and $N = 16$) to get approximations of the discretization errors, which may be used to give an indication of the order of the accuracy of the solution. The measure that we use is the discrete energy norm, defined by

$$|||D^h||| = \langle L^h D^h, D^h \rangle^{\frac{1}{2}}, \tag{V.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, L^h is the operator defined in Equation IV.2, and $D^h = T^h - T^*$ approximates the discretization error, where T^h is the solution to Equation IV.2 on grid h, and T^* is the "exact" solution (i.e., solution of Equation IV.2 with $N = 64$ sampled on grid h). (see [Ref. 11])

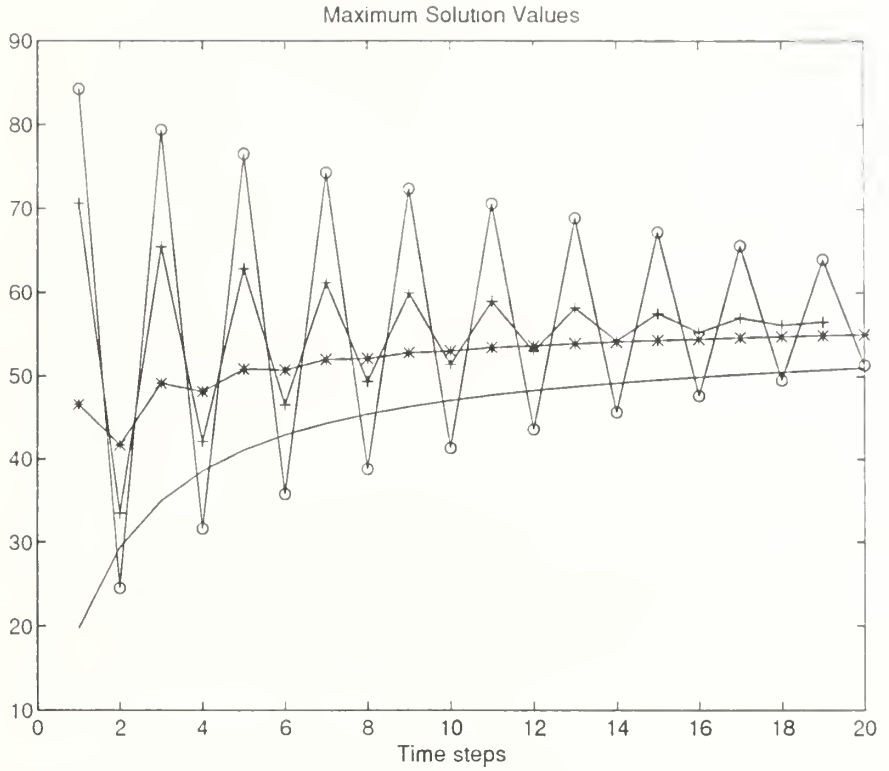

Figure 22. The maximum values of the solution for each time step, where $\alpha = \frac{1}{2}$; "o" indicates $g = h$, "+" indicates $g = h^2$, "*" indicates $g = h^{\frac{5}{2}}$, "–" indicates $g = h^3$. Care must be taken in analyzing the figure since the apparent equivalence of time steps can be misleading, e.g., for $g = h^2$ and $h = \frac{1}{16}$, it takes 16 time steps to "equal" one time step for $g = h$.

The solutions for an inital time step[1], for a time of one second, and for steady state have been compared using the discrete energy norm, with the resulting discretization errors indicated in Table VII. We might hope to see a common factor of decrease as we move to finer grids. That is, if the discretization error on grid h were of the order $e = ch^p$, for some constant c, then the ratio of errors for grids h and $\frac{h}{2}$

[1] With $g = h$, the size of a time step differs with grid size. We have adjusted for this difference by requiring that the same "amount" of time elapse for solutions used in the comparison, e.g., for the initial time step, the "exact" solution on $N = 64$ after eight time steps is compared to the solution on $N = 32$ after four time steps, to the solution on $N = 16$ after two time steps, and to the solution on $N = 8$ after one time step.

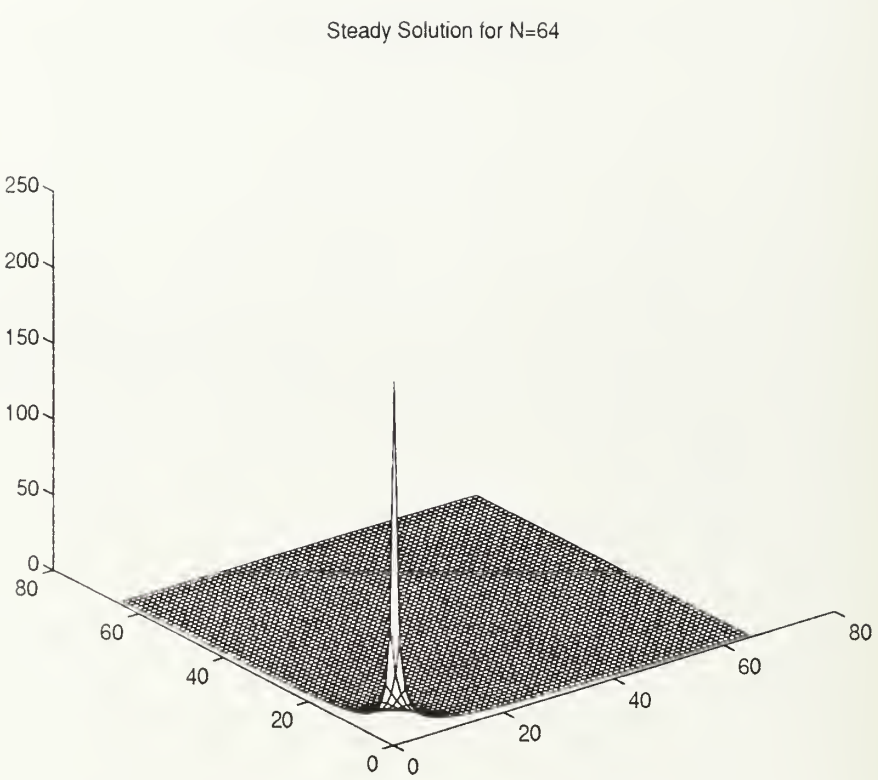

Figure 23. The steady state "exact" solution for $N = 64$.

would be

$$\left|\frac{e_{\frac{h}{2}}}{e_h}\right| = \left|\frac{c(\frac{h}{2})^p}{ch^p}\right| = \left|2^{-p}\right|.$$

Thus, a decrease of a factor of four would indicate that the process achieves $O(h^2)$ discretization error (see [Ref. 11]). Table VII indicates that the discretization error decreases by a factor of about two when moving from grid size $N = 8$ to $N = 16$, and by a factor of about four when moving from grid $N = 16$ to $N = 32$. A grid size $N = 8$ may be too small to adequately reflect the accuracy of the solution, which would leave the factor of four, possibly suggesting that the discretization error is $O(h^2)$. Even so, a specific estimate of the discretization error is best deferred until more information can be obtained, i.e., solutions on finer grids.

	$N = 8$	$N = 16$	$N = 32$
Initial time step	43.07	24.02	6.85
Time = "1 second"	42.99	23.98	6.87
Steady state	42.99	23.98	6.87

Table VII. Discretization Errors for the Iterative Solution.

VI. MULTILEVEL SOLUTION

Now that we can solve a discrete representation of the heat equation (Equation IV.2) iteratively, we want to apply multigrid techniques in order to attempt to accelerate the solution process. Multigrid is a method to improve on solution by relaxation by making use of the advantages of working on successively coarser grids (for a more complete treatment, see [Ref. 3]). It has been used with marked success in solving a variety of problems, specifically elliptic partial differential equations. Although the problem we are solving is parabolic, in the time-stepping regime we must solve an elliptic problem at each time step. Thus, multigrid may prove useful in streamlining our solution process. We begin by introducing the multigrid method and some of the exigencies of the use of multiple grids. We then analyze some of the characteristics of the method using local mode analysis, and present numerical results from the solution process. The amount of work to achieve the iterative solution serves as a baseline against which the computational work for the multilevel solution is compared.

As indicated in the analysis of the Gauss-Seidel and line relaxation schemes in Chapter IV, the high frequency (oscillatory) components of the error are extirpated much more efficiently by these schemes than are the low frequency (smooth) components. The result of applying a relaxation scheme to generate an approximate solution on a specific grid size (call it a fine grid, size N), is that the oscillatory components of the error are smoothed. After sufficient work has been done on the fine grid to smooth the error (in effect, when the relaxation process stalls), the problem may be shifted to a coarser grid (grid size $\frac{N}{2}$) where the smooth components of the error become oscillatory (see figure 24, taken from [Ref. 3]). The relaxation scheme is then applied again to smooth the oscillatory components of the error. The information gained from smoothing on the coarse grid is transferred back to the fine grid, where it becomes a correction to the original approximation. That is, the result of smoothing

on the fine grid, transferring to the coarse grid, smoothing on the coarse grid, and transferring back to the fine grid is to produce a **coarse grid correction**. This process of using multiple grids to obtain an approximate solution has the advantages of a) more effectively targeting the error components, and b) requiring less work, since the coarse grid has only 2^{-d} the number of points on the fine grid, where d is the dimension of the domain. Multigrid also allows for ways to improve upon the initial guess that is used in the smoothing (see [Ref. 3]). However, since we are solving a time-stepping problem, where the current solution becomes the initial guess for the next time step, this will not be a concern for us. Moreover, while there is a good deal more to multigrid than coarse grid correction, this concept forms the foundation of our solution process.

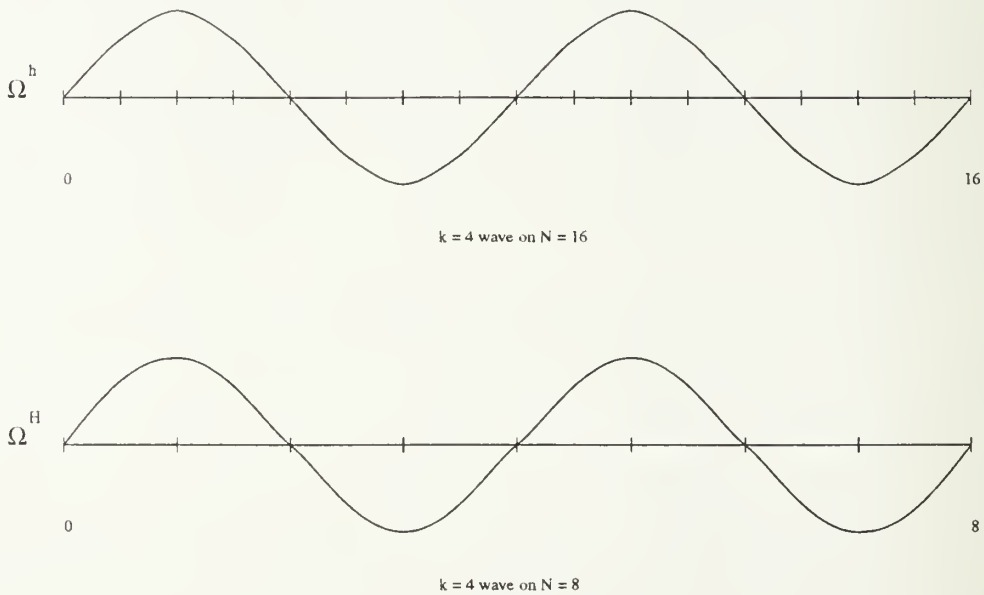

Figure 24. A wave with wave number $k = 4$ on Ω^h ($N = 16$) is projected onto Ω^H ($N = 8$). For $N = 16$, $k = 4$ implies that the wave is $\frac{4}{16} = \frac{1}{4}$ the way up the spectrum, while, for $N = 8$, $k = 4$ is the wave that is $\frac{4}{8} = \frac{1}{2}$ way up the spectrum. Thus, the coarse grid "sees" a wave which is more oscillatory.

In order to make effective use of this technique, we must specify what information to transfer, and how to transfer it; first we consider what information to transfer.

Recall that we are attempting to solve Equation IV.3.

$$\mathrm{M}\vec{T} = \vec{f},$$

and suppose that we have an approximate solution $\vec{v}$; the error is given by $\vec{e} = \vec{T}^* - \vec{v}$, where $\vec{T}^*$ is the exact solution. Thus, the error satisfies

$$\begin{aligned} \mathrm{M}\vec{e} &= \vec{f} - \mathrm{M}\vec{v} \\ &\equiv \vec{r}, \end{aligned} \tag{VI.1}$$

where $\vec{r}$ is the residual. The smooth components of the error are the troublesome ones, since the relaxation schemes "kill" the oscillatory modes. Since smooth modes on a fine grid become oscillatory when projected onto a coarse grid, it is natural to consider transferring a representation of the error, i.e., the residual, from one grid to the next[1]. In this way, the relaxation schemes used on the coarse grid will reduce components of the error that could not be reduced on the fine grid. Additionally, we know that relaxation smooths the error, and therefore we can *accurately* represent the error on the coarse grid. After transferring the residual to the coarse grid, we can relax on the coarse grid version of the residual equation (Equation VI.1), solving for the error. Thus, when we have determined an approximation to the error on the coarse grid, we can transfer it back to the fine grid as a correction to the current approximation. That is, since $\vec{T}^* = \vec{v} + \vec{e}$, if we know $\vec{v}$ we can correct it by adding an approximation of $\vec{e}$. This process can be outlined as in the following steps, where h represents the grid spacing on the fine grid, H the spacing on the coarse grid, and we assume $2h = H$.

Coarse grid correction (two-grid scheme)

- Relax on $\mathrm{M}^h\vec{T}^h = \vec{f}^h$ on Ω^h to obtain an approximation $\vec{v}^h$.

[1]There are a number of other good reasons to transfer the residual, as outlined in [Ref. 3]. However, there are other multigrid schemes that do not transfer the residual, such as the Full Approximation Scheme (FAS) (see [Ref. 2] and [Ref. 1]). FAS is useful in dealing with nonlinear problems but, since our problem is linear, we will not employ it.

- Compute the residual $\vec{r}^h = \vec{f}^h - \mathbf{M}^h \vec{v}^h$.
- Solve the residual equation $\mathbf{M}^H \vec{e}^H = \vec{r}^H$ on Ω^H to obtain an approximation to the error $\vec{e}^H$.
- Correct the approximation obtained on Ω^h with the error estimate obtained on Ω^H: $\vec{v}^h \leftarrow \vec{v}^h + \vec{e}^H$.[Ref. 3]

The task of solving on Ω^h can, thus, be exchanged for the task of relaxing on Ω^h, and then solving on Ω^H. The requirement to solve on Ω^H can be treated in like fashion; the task of solving can be exchanged for the task of relaxing, and then solving on the next coarser grid. This procdeure of transferring to successively coarser grids can be continued until a grid is reached on which an "exact" solution is possible. That is, the solution can be generated by recursive use of the coarse grid correction, which can be represented as follows:

Solve on Ω^h by relaxing on Ω^h and solving on Ω^{2h}.
Solve on Ω^{2h} by relaxing on Ω^{2h} and solving on Ω^{4h}.
Solve on Ω^{4h} by relaxing on Ω^{4h} and solving on Ω^{8h}.
$\vdots$
"Exact" solve on coarsest grid.
$\vdots$
Correct on Ω^{4h}.
Correct on Ω^{2h}.
Correct on Ω^h.

This process of starting on a fine grid, moving to the coarsest grid, and then going back to the fine grid is known as a **V-cycle** (see Figure 25). Each time the transfer to a coarser grid is made, the relaxation process there smooths another portion of the error component spectrum (see Figure 26). Thus, by the time the coarsest grid is reached and the problem has been solved, all of the error components have been smoothed. When the problem is solved on the coarsest grid, the approximate error is transferred to the next finer grid to become a correction. The corrected error is then passed to the next finer grid, and so on until we are back on the finest grid,

where the original approximation is updated with the composite correction from the coarser grids. While there are other multigrid methods which may prove more useful, depending on the nature of the problem (see [Ref. 3]), we use the V-cycle.

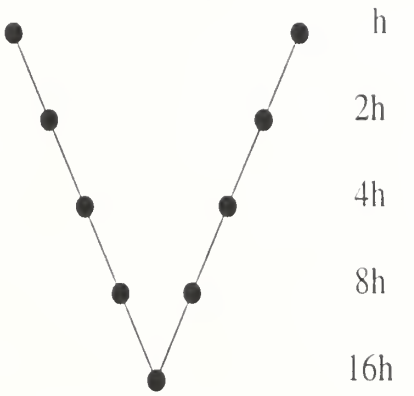

Figure 25. Schedule of grids for a V-cycle.

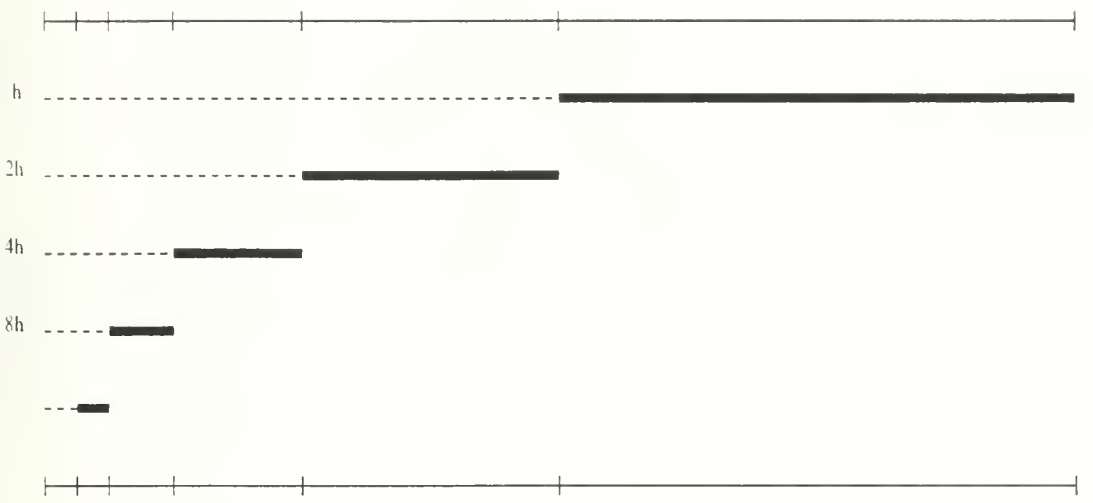

Figure 26. Error component spectrum for various grid sizes. On the fine grid, h, relaxation smooths the high frequency components of the error (heavy black line). When the problem is transferred to the next coarser grid, $2h$, the high frequencies of the remaining (unsmoothed) components (heavy black line) are smoothed. This process continues all the way down to the coarsest grid, where the problem is solved, eliminating the remaining components of the error. By continuing to transfer to successively coarser grids, all frequency components of the error are smoothed.

So far, we have not indicated how information will transferred, nor how an error estimate on the coarse grid (solving $\mathbf{M}^H \vec{e}^H = \vec{r}^H$) will be computed. We will

deal with the former question in the next section; the latter question revolves around deciding what to use as the coarse grid operator, $\mathbf{M}^H$. One approach is to simply discretize the problem on the coarse grid in precisely the same fashion that it is discretized on the fine grid. This has the advantages that the work has already been done, and that it is easy to implement computationally. While this method generally works (on model problems), it has the disadvantage that it is often not true to the physics, e.g., conservation may no longer be enforced. There are other methods for determining the coarse grid operator which are true to the physics and allow for strong theoretical treatment of convergence and other properties (see [Ref. 2] and [Ref. 3]). However, because discretization on the coarse grid generally works, and because it simplifies the computations, forming $\mathbf{M}^H$ in this manner is the method that we use.

We conclude this section with a discussion of what happens to boundary conditions when we transfer the residual. The boundary conditions for our problem are a mixture of Dirichlet and Neumann conditions (Equations II.18, II.19, and II.20). By relaxing on Equation IV.2,

$$L[T] = f,$$

(where we have dropped the superscript h) we obtain an approximation v, which gives rise to the residual equation, $r = f - L[v]$. Suppose that T^* is the exact solution, so that $f = L[T^*]$; we require that the approximation satisfy the same conditions as the exact solution on the boundary, $\partial\Omega$. Thus, the residual equation becomes

$$\begin{aligned} r &= L[T^*] - L[v] \\ &= [\int_V (\frac{1}{g} - \alpha\kappa\nabla^2)T^* dV] - [\int_V (\frac{1}{g} - \alpha\kappa\nabla^2)v dV] \\ &= \frac{1}{g}\int_V (T^* - v)dV - \alpha\kappa\int_V (\nabla^2 T^* - \nabla^2 v)dV \\ &= \frac{1}{g}\int_V (T^* - v)dV - \alpha\kappa\int_S (\nabla T^* - \nabla v)\cdot\hat{n}dS. \end{aligned} \qquad \text{(VI.2)}$$

Therefore, since the boundary conditions satisfy $(T^* = v)|_{\partial\Omega}$ and $(\frac{\partial T^*}{\partial n} = \frac{\partial v}{\partial n})|_{\partial\Omega}$[2], all

[2] $\frac{\partial}{\partial n} = \nabla\cdot\hat{n}$ denotes the outward normal on $\partial\Omega$.

boundary conditions for r become homogeneous; the first term on the right hand side of Equation VI.2 indicates that this is true for the Dirichlet conditions, the second term indicates that it is true for the Neumann conditions. Thus, since we are solving the residual equation on all the successively coarser grids, we impose homogeneous boundary conditions on all but the finest grid.

A. INTERGRID TRANSFERS

In order to transfer information between grids, we develop operators that **restrict** data from a fine grid to a coarse grid, and **interpolate** data from a coarse grid to a fine grid (see [Ref. 3]). These operators are designated by $\mathbf{I}_{\text{subscript}}^{\text{superscript}}$, where the subscript indicates the grid from which the information is being transferred, and the superscript indicates the grid to which the information is being transferred. Thus, the **restriction operator** is indicated by $\mathbf{I}_h^H$, since the information goes from the fine grid to the coarse grid, and the **interpolation operator** is represented by $\mathbf{I}_H^h$, since the information goes from the coarse grid to the fine grid. In other words, restriction is a process of determining what values to assign to coarse grid points, based on the values at fine grid points; interpolation is a process of determining what values to assign to fine grid points, based on the values at coarse grid points. In [Ref. 1], such operators are developed which take advantage of FVE characteristics. We consider two types of restriction operators, one which follows from the physics and one which is very simple, and one type of interpolation operator.

One of the advantages of the FVE method is its fidelity to conservation considerations. The restriction technique that we describe first follows from the conservation notion. In order to determine what value to assign to a coarse grid point, the fine grid is laid over the coarse grid, where the coarse grid control volumes align with fine grid lines (see Figure 27). The control volume for a given coarse grid point includes all or part of the control volumes of nine fine grid points. Hence, the value of a quantity on the coarse grid includes contributions from the value of the quantity on each of the

nine fine grid points. For each of these fine grid points, the contribution is determined by computing the fraction of the associated fine grid volume that is contained in the coarse grid volume. The value of the quantity at the fine grid point is multiplied by this fraction, and the result is the contribution to the coarse-grid value. The value of the quantity at the coarse grid point is thus the sum of the nine contributions determined in this manner.

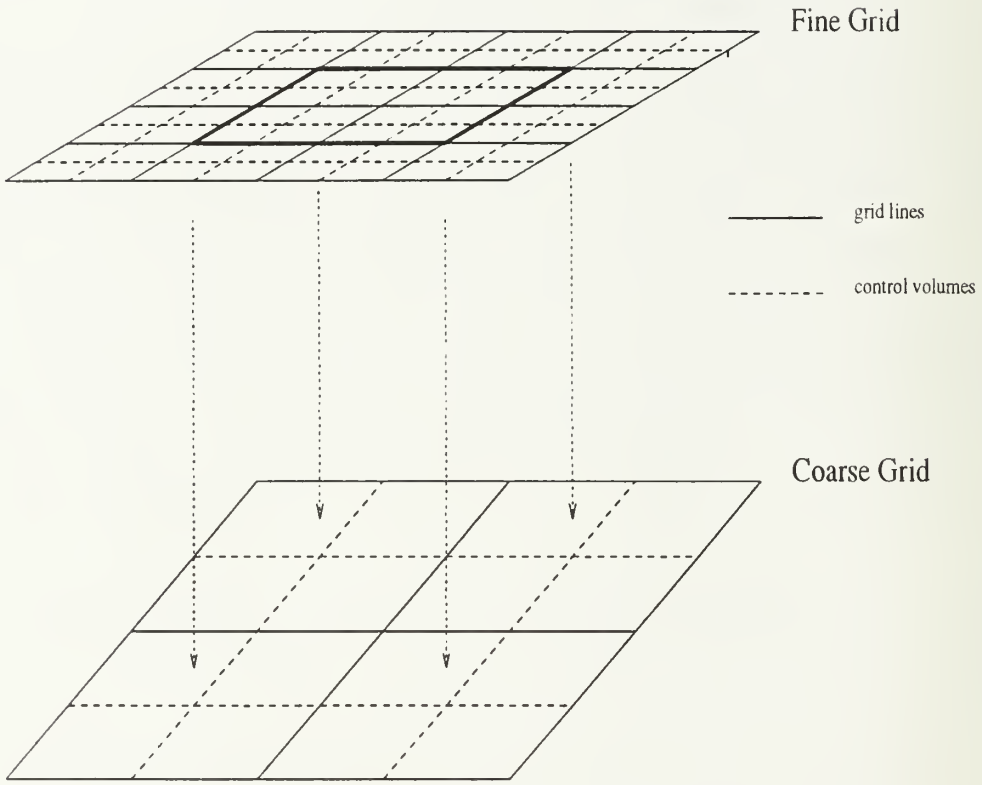

Figure 27. The "conservation restriction" operator.

In the case of cylindrical coordinates, care must be taken when deciding what percentage of the fine grid volumes fall within each coarse grid volume. Since the volumes change with the radius, the restriction operator will be a function of grid location. The stencil for the restriction operator gives the percentage of the fine grid value that is assigned to the coarse grid value based on the percentage of the fine grid volume that falls within the coarse grid volume (see Figure 28). For example, the amount of the value of the point at $(k, j+1)$ that is used in the sum is found by

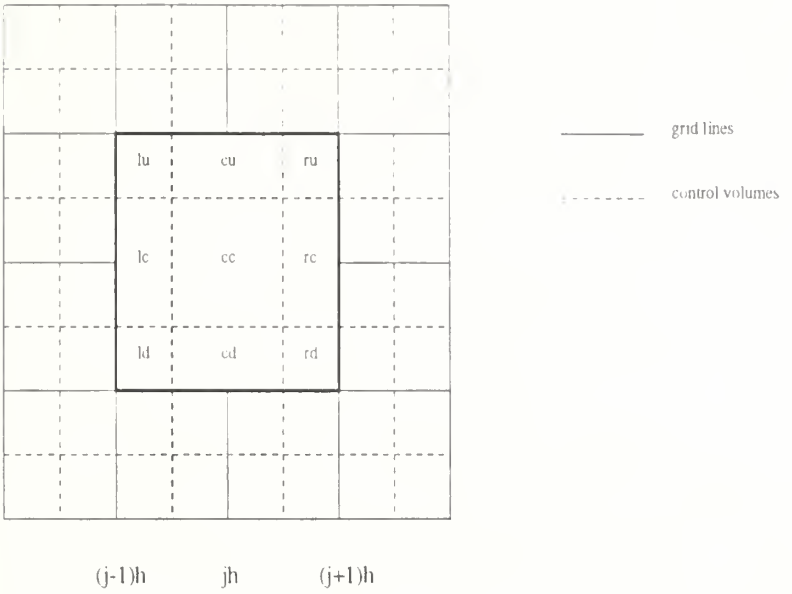

Figure 28. The fine grid region, comprising contributions from the nine fine grid control volumes, that corresponds to a coarse grid control volume. The central fine grid point is labeled (k, j), (k, j even) for which the radial distance $r = jh$, and corresponds to coarse grid point $(\frac{k}{2}, \frac{j}{2})$.

first determining how much of its fine grid control volume (right-center, or rc) falls within the region that corresponds to the coarse grid control volume. This volume is then divided by the total control volume for the point $(k, j+1)$ to determine the volume percentage. So, if the fine grid location is at the radial distance $r = jh$, then we have for the volume rc (a toroidal prism)

$$\pi h[(j+1)^2h^2 - (j+\frac{1}{2})^2h^2)],$$

or

$$\pi h^3(j+\frac{3}{4}),$$

which is divided by the fine grid control volume for the point at $(k, j+1)$,

$$\pi h((j+\frac{3}{2})^2h^2 - (j+\frac{1}{2})^2h^2)),$$

or

$$\pi h^3(2j+2),$$

giving

$$\frac{j+\frac{3}{4}}{2(j+1)}.$$

Similarly, the ratio for the left-center (lc) portion of the stencil is

$$\frac{j-\frac{3}{4}}{2(j-1)},$$

and the remainder of the stencil is given by,

$$\begin{bmatrix} \frac{1}{4}(\frac{j-\frac{3}{4}}{j-1}) & \frac{1}{2} & \frac{1}{4}(\frac{j+\frac{3}{4}}{j+1}) \\ \frac{1}{2}(\frac{j-\frac{3}{4}}{j-1}) & 1 & \frac{1}{2}(\frac{j+\frac{3}{4}}{j+1}) \\ \frac{1}{4}(\frac{j-\frac{3}{4}}{j-1}) & \frac{1}{2} & \frac{1}{4}(\frac{j+\frac{3}{4}}{j+1}) \end{bmatrix} T^h_{k,j}.$$

This is a stencil for interior points; similar calculations determine the stencils for boundary points.

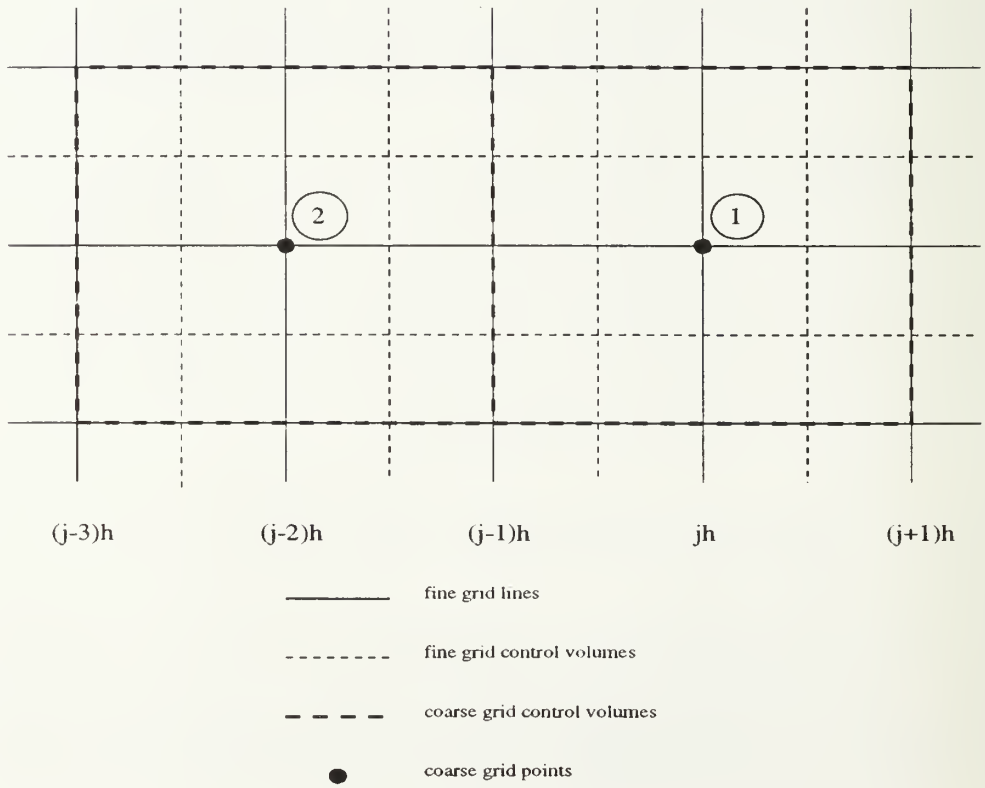

Figure 29. Neighboring coarse grid points.

Since this operator is both somewhat out of the ordinary and dependent on grid location, we consider it worthwhile to try to get a sense of how this restriction operator

treats the quantities being restricted. For this purpose, we consider a comparison between the values assigned to neighboring coarse grid points, as in Figure 29, when the restriction operator acts on a grid with constant value $T_{k,j} = 1$. The value for coarse grid point number 1 is given by

$$\frac{j-\frac{3}{4}}{j-1} + 2 + \frac{j+\frac{3}{4}}{j+1},$$

or

$$\begin{aligned} 2 + 2\left(\frac{j^2-\frac{3}{4}}{(j+1)(j-1)}\right) &= 2 + 2\left(\frac{j^2-1+\frac{1}{4}}{(j+1)(j-1)}\right) \\ &= 4 + \frac{1}{2(j+1)(j-1)}. \end{aligned}$$

The corresponding value for coarse grid point number 2 is

$$\frac{j-\frac{11}{4}}{j-3} + 2 + \frac{j-\frac{5}{4}}{j-1}$$

or

$$\begin{aligned} 2 + 2\left(\frac{j^2-4j+\frac{13}{4}}{(j-3)(j-1)}\right. &= 2 + 2\left(\frac{j^2-4j+3+\frac{1}{4}}{(j-3)(j-1)}\right) \\ &= 4 + \frac{1}{2(j-3)(j-1)}. \end{aligned}$$

Since $(j+1) > (j-3)$ for $2 \le j \le N-2$, then $\frac{1}{j+1} < \frac{1}{j-3}$, which implies

$$\frac{1}{2(j+1)(j-1)} < \frac{1}{2(j-3)(j-1)},$$

and therefore

$$4 + \frac{1}{2(j+1)(j-1)} < 4 + \frac{1}{2(j-3)(j-1)}.$$

Thus, if the restriction operator acts on a constant vector, the values assigned to the coarse grid points decrease as the radial distance increases.

Does this make sense in terms of the physics of the problem? Should a constant function remain a constant after restriction, or not? In general, one might expect that restriction would transfer a constant to a constant. The conservaton restriction

operator acts on the basis of volume percentages and, thus, considers the *values* assigned to each control volume in terms of the actual *volume*. In some sense, this restriction operator converts the values assigned to a control volume into a "density", and then transfers this amount. In the case of cylindrical coordinates, assigning a constant value to each control volume means that the ratio of the value to its volume decreases with radial distance. Thus, the result of the above computation is no surprise. Moreover, it seems to make sense physically to transfer "densities" in terms of enforcing conservation. However, in order to regain the fine-grid interpretation of the information after it has been transferred may require more work. In particular, some compensation may need to be made relative to the coarse grid volumes (i.e., convert the "density" back to the original type of information). The question of how or whether to make this compensation will not be addressed here. Additionally, it is certainly possible that the information to be transferred does not make sense in the context of densities. In that case, physical intuition may need to be suspended while the efficacy of the solution process is determined.

The other restriction operator we introduce is the injection operator. Injection is accomplished by simply assigning values to the coarse grid points directly from the corresponding fine grid points, $v^H_{k,j} = v^h_{2k,2j}$ (see [Ref. 3]). The injection operator has the advantage of being a linear operator, whereas the conservation restriction operator may not be linear. Note that, when the conservation restriction operator is applied to a grid with constant value $T_{k,j} = 1$, the result is $T_{2k,2j} \approx 4$. If the injection restriction operator is applied to the same grid, $T_{2k,2j} = 1$. Thus, care must be taken when comparing these two restriction operators.

In order to transfer from the coarse grid to the fine grid, we must choose an interpolation operator. Since we are assuming that the basis functions for the FVE method are linear, a natural choice is linear interpolation. A continuous H-piecewise linear function is also a continuous h-piecewise linear function. That is, the "coarse" finite element space is a subspace of the "fine" finite element space (see [Ref. 1]).

This means that linear interpolation corresponds to the "natural embedding" from the coarse finite element space to the fine element space. Its advantages are its simplicity and its linearity. The stencil is given by

$$\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} T^H_{k,j}.$$

B. AMPLIFICATION MATRIX

Now that we have our fine grid operators, coarse grid operators, and intergrid transfer operators, we conduct analysis using the DFT (see Appendix A) similar to that in Chapter IV to obtain some indication as to how a two-level multigrid scheme using these operators will perform. We start by noting that the process of relaxing on the fine grid, obtaining a coarse grid correction, and relaxing again on the fine grid can be represented by the two-level error propagation matrix, $P^{\nu_2}(CG)P^{\nu_1}$. Note that, while h as a superscript identifies grid spacing, ν_1 and ν_2 are exponents. Thus, P^{ν_1} and P^{ν_2} represent relaxation ν_1 and ν_2 times on the fine grid, and CG is the coarse grid correction. Recall that coarse grid correction is the following series of steps:

Coarse grid correction (two-grid scheme)

- Relax on $L^h T^h = f^h$ on Ω^h to obtain an approximation v^h.
- Compute the residual $r^h = f^h - L^h v^h$.
- Restrict the residual $r^H = I^H_h r^h$.
- Solve the residual equation $L^H e^H = r^H$ on Ω^H to obtain an approximation to the error e^H.
- Interpolate the error $e^h = I^H_h e^H$.
- Correct the approximation obtained on Ω^h with the error estimate obtained on Ω^H: $v^h \leftarrow v^h + e^h$.

Thus, CG can be written as

$$CG = I - I_H^h (L^H)^{-1} I_h^H L^h,$$

where I is the identity operator, I_h^H and I_H^h are the restriction and interpolation operators, L^h is the fine grid operator, and $(L^H)^{-1}$ is the coarse grid smoother. Each of the operators in the two-level error propagation matrix is expanded in a DFT (see Appendix A and Chapter IV), making the resulting expansions functions of $l, m = -\frac{N}{2} + 1 : \frac{N}{2}$, the indices of the expansion in the frequency domain. These expansions are used to construct an **amplification matrix**, $\mathbf{A}(l, m)$, which is also a function of l and m. The largest spectral radius of this matrix over the values that l, m have on the coarse grid, i.e., $l, m = -\frac{N}{4} + 1 : \frac{N}{4}$ (this relationship is discussed later), will give us a two-level **asymptotic convergence factor**, $\bar{\lambda}$. This factor, much like the values determined for the relaxation schemes, will give a bound on how well the two-level scheme can be expected to perform. In order to guarantee convergence, $\bar{\lambda} < 1$ is necessary; the smaller $\bar{\lambda}$ is, the better.

In Chapter IV, we used DFT expansions of the error equations derived from the relaxation operators to determine the frequency domain coefficients for the operators. We now use the same technique, again using error equations, to determine the matrix entries for the operators in the two-level scheme. The amplification matrix is formed by multiplying together these operator matrices,

$$\mathbf{A}(l, m) = (\breve{\mathbf{P}}^h)^{\nu_2} (\mathbf{I} - \breve{\mathbf{I}}_H^h (\breve{\mathbf{L}}^H)^{-1} \breve{\mathbf{I}}_h^H \breve{\mathbf{L}}^h)(\breve{\mathbf{P}}^h)^{\nu_1}, \tag{VI.3}$$

where

- $\breve{\mathbf{L}}^h$ is the 4x4 matrix for the fine grid discretization operator.
- $\breve{\mathbf{P}}^h$ is the 4x4 matrix for the fine grid relaxation operator.
- $(\breve{\mathbf{L}}^H)^{-1}$ is the 1x1 matrix for the coarse grid relaxation operator.
- $\breve{\mathbf{I}}_h^H$ is the 1x4 matrix for the restriction operator.
- $\breve{\mathbf{I}}_H^h$ is the 4x1 matrix for the interpolation operator.

- I is the 4x4 identity matrix. (see [Ref. 2])

We go through the details of computing the matrix entries for the fine grid operator, L^h; results for the remaining operators are then presented.

From the FVE stencils for L^h, we have the operator equation

$$v_{2k,2j}^{h(new)} = L^h v_{2k,2j}^{h(old)},$$

or

$$v_{2k,2j}^{h(new)} = \frac{h^3}{g384}\begin{pmatrix} & (64j-5)v_{2k+1,2j}^{h(old)} & +(32j+5)v_{2k+1,2j+1}^{h(old)} \\ +(64j-11)v_{2k,2j-1}^{h(old)} & +448jv_{2k,2j}^{h(old)} & +(64j+11)v_{2k,2j+1}^{h(old)} \\ +(32j-5)v_{2k-1,2j-1}^{h(old)} & +(64j+5)v_{2k-1,2j}^{h(old)} & \end{pmatrix}$$

$$-\frac{\alpha\kappa h}{2}\begin{pmatrix} & 4jv_{2k+1,2j}^{h(old)} & \\ +(-1+4j)v_{2k,2j-1}^{h(old)} & -16jv_{2k,2j}^{h(old)} & +(1+4j)v_{2k,2j+1}^{h(old)} \\ & +4jv_{2k-1,2j}^{h(old)} & \end{pmatrix},$$

where the superscript h indicates a vector component on the fine grid, (old) and (new) indicate before and after the operator acts, and subscripts $2k$ and $2j$ are the respective fine grid row and column indices. These indicial values are used because we shortly will need to discuss the correspondence between the fine and coarse grid points; only those fine grid points with indices $2k$ and $2j$ correspond to coarse grid points (with indices j and k). Ordinarily (see [Ref. 2]), the consideration of indicial values is not a concern. In this case, however, since the stencils are grid location dependent, and since the analysis of the amplification matrix must apply to both the fine and coarse grids, the j that is used is the column index on the coarse grid, requiring that $2j$ be the column index on the fine grid. Thus, expanding both sides in a DFT, we have

$$\sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}}\sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}} V_{l,m}^{(n)}C(l,m) = \sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}}\sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}}\Bigg[\frac{h^3}{g384}\Big((64j-5)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi l}{N}}$$
$$+(32j+5)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi(l+m)}{N}}$$
$$+(64j-11)V_{l,m}^{(o)}C(l,m)e^{\frac{-\imath 2\pi m}{N}}$$
$$+448jV_{l,m}^{(o)}C(l,m)+(64j+11)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi m}{N}}$$
$$+(32j-5)V_{l,m}^{(o)}C(l,m)e^{-\frac{\imath 2\pi(l+m)}{N}}$$
$$+(64j+5)V_{l,m}^{(o)}C(l,m)e^{-\frac{\imath 2\pi l}{N}}\Big)$$
$$-\frac{\alpha\kappa h}{2}\Big(4jV_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi l}{N}}$$
$$+(-1+4j)V_{l,m}^{(o)}C(l,m)e^{-\frac{\imath 2\pi m}{N}}$$
$$-16jV_{l,m}^{(o)}C(l,m)+(1+4j)V_{l,m}^{(o)}C(l,m)e^{\frac{\imath 2\pi m}{N}}$$
$$+4jV_{l,m}^{(o)}C(l,m)e^{\frac{-\imath 2\pi l}{N}}\Big)\Bigg],$$

where $2 \leq 2j \leq N-2$ on a grid labeled $0:N$, since we are in the interior, $l,m = -\frac{N}{2}+1 : \frac{N}{2}$, and

$$C(l,m) = e^{\frac{\imath 4\pi kl}{N}}e^{\frac{\imath 4\pi jm}{N}}.$$

When we transfer information from the fine grid to the coarse grid, we must remember that not all of the freqencies on the fine grid can be represented on the coarse grid. The highest frequency that can be resolved on any grid is the **Nyquist frequency** (see [Ref. 12]), $f = \frac{1}{2\Delta x}$, where Δx is the grid spacing. Hence, on the coarse grid, Ω^H, $\Delta x = \frac{2}{N}$ and $f = \frac{N}{4}$. Therefore, in order to make this analysis germane to both fine and coarse grids, we must rewrite the sums in the expansions so that $|l|, |m| \leq \frac{N}{4}$. This can be done as follows:

$$\sum_{l=-\frac{N}{2}+1}^{\frac{N}{2}}\sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}} V_{l,m}C(l,m) = \sum_{l=-\frac{N}{4}+1}^{\frac{N}{4}}\sum_{m=-\frac{N}{4}+1}^{\frac{N}{4}}[V_{l,m}C(l,m) \tag{VI.4}$$
$$+V_{l+\frac{N}{2},m}C(l+\frac{N}{2},m)+V_{l,m+\frac{N}{2}}C(l,m+\frac{N}{2})+V_{l+\frac{N}{2},m+\frac{N}{2}}C(l+\frac{N}{2},m+\frac{N}{2})\Big].$$

Thus, rewriting the DFT expansion of the operator equation to consider those frequencies that can be represented on the coarse grid and, making use of the orthogonality property of the complex exponential (see Appendix A), we can equate individual terms of the sum, and then divide by $C'(l,m) \neq 0$ to give

$$\begin{aligned} V_{l,m}^{(n)} &= \left[\frac{h^3}{g384}\left((64j-5)V_{l,m}^{(o)}e^{\frac{\imath 2\pi l}{N}} + (32j+5)V_{l,m}^{(o)}e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\ &\quad +(64j-11)V_{l,m}^{(o)}e^{\frac{-\imath 2\pi m}{N}} + 448jV_{l,m}^{(o)} + (64j+11)V_{l,m}^{(o)}e^{\frac{\imath 2\pi m}{N}} \\ &\quad \left.+(32j-5)V_{l,m}^{(o)}e^{-\frac{\imath 2\pi(l+m)}{N}} + (64j+5)V_{l,m}^{(o)}e^{-\frac{\imath 2\pi l}{N}}\right) \\ &\quad -\frac{\alpha\kappa h}{2}\left(4jV_{l,m}^{(o)}e^{\frac{\imath 2\pi l}{N}} + (-1+4j)V_{l,m}^{(o)}e^{-\frac{\imath 2\pi m}{N}}\right. \\ &\quad \left.\left.-16jV_{l,m}^{(o)} + (1+4j)V_{l,m}^{(o)}e^{\frac{\imath 2\pi m}{N}} + 4jV_{l,m}^{(o)}e^{\frac{-\imath 2\pi l}{N}}\right)\right], \end{aligned}$$

or, factoring out $V_{l,m}^{(o)}$,

$$\begin{aligned} V_{l,m}^{(n)} &= \left[\frac{h^3}{g384}\left((64j-5)e^{\frac{\imath 2\pi l}{N}} + (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\ &\quad +(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j + (64j+11)e^{\frac{\imath 2\pi m}{N}} \\ &\quad \left.+(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} + (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\ &\quad -\frac{\alpha\kappa h}{2}(4je^{\frac{\imath 2\pi l}{N}} + (-1+4j)e^{-\frac{\imath 2\pi m}{N}} \\ &\quad \left.-16j + (1+4j)e^{\frac{\imath 2\pi m}{N}} + 4je^{\frac{-\imath 2\pi l}{N}}\right)\Big] V_{l,m}^{(o)}, \end{aligned}$$

and

$$\begin{aligned} V_{l+\frac{N}{2},m}^{(n)} &= \left[\frac{h^3}{g384}\left(-(64j-5)e^{\frac{\imath 2\pi l}{N}} - (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\ &\quad +(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j + (64j+11)e^{\frac{\imath 2\pi m}{N}} \\ &\quad \left.-(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} - (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\ &\quad -\frac{\alpha\kappa h}{2}\left(-4je^{\frac{\imath 2\pi l}{N}} + (-1+4j)e^{-\frac{\imath 2\pi m}{N}}\right. \\ &\quad \left.\left.-16j + (1+4j)e^{\frac{\imath 2\pi m}{N}} - 4je^{\frac{-\imath 2\pi l}{N}}\right)\right] V_{l+\frac{N}{2},m}^{(o)}, \end{aligned}$$

$$
\begin{aligned}
V^{(n)}_{l,m+\frac{N}{2}} \;=\; & \left[\frac{h^3}{g384}\left((64j-5)e^{\frac{\imath 2\pi l}{N}} - (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\
& -(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j - (32j+11)e^{\frac{\imath 2\pi m}{N}} \\
& \left.-(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} + (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\
& -\frac{\alpha\kappa h}{2}(4je^{\frac{\imath 2\pi l}{N}} - (-1+4j)e^{-\frac{\imath 2\pi m}{N}} \\
& \left.-16j-(1+2j)e^{\frac{\imath 2\pi m}{N}} + 4je^{\frac{-\imath 2\pi l}{N}}\right)\Big] V^{(o)}_{l,m+\frac{N}{2}},
\end{aligned}
$$

and

$$
\begin{aligned}
V^{(n)}_{l+\frac{N}{2},m+\frac{N}{2}} \;=\; & \left[\frac{h^3}{g384}\left(-(64j-5)e^{\frac{\imath 2\pi l}{N}} + (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\
& -(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j - (64j+11)e^{\frac{\imath 2\pi m}{N}} \\
& \left.+(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} - (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\
& -\frac{\alpha\kappa h}{2}(-4je^{\frac{\imath 2\pi l}{N}} - (-1+4j)e^{-\frac{\imath 2\pi m}{N}} \\
& \left.-16j-(1+4j)e^{\frac{\imath 2\pi m}{N}} - 4je^{\frac{-\imath 2\pi l}{N}}\right)\Big] V^{(o)}_{l+\frac{N}{2},m+\frac{N}{2}},
\end{aligned}
$$

since

$$
e^{\frac{\imath 2\pi(l+\frac{N}{2})}{N}} = e^{\frac{\imath 2\pi l}{N}}e^{i\pi} = -e^{\frac{\imath 2\pi l}{N}}, \quad \text{and} \quad e^{\frac{\imath 2\pi(m+\frac{N}{2})}{N}} = -e^{\frac{\imath 2\pi m}{N}}.
$$

The matrix of coefficients for the fine grid operator is a 4x4 diagonal matrix, $\check{\mathrm{L}}^h = (\Lambda_{i,i})$ (see [Ref. 2]). To see why the matrix is 4x4, recall that the sum in the original DFT expansion of the operator equation was rewritten so that the indices satisfy $|l|, |m| \leq \frac{N}{4}$. One result of this is to rewrite each of the individual terms of the sum as a combination four terms (Equation VI.4). That is, for each dimension of the problem, represented by l and m, there are two components of each individual term in the sum, which are indexed by $|l|, |m| \leq \frac{N}{4}$ and $\frac{N}{4} \leq |l|, |m| \leq \frac{N}{2}$. Thus, the matrix of coefficients is a 2^2x2^2 matrix, where each row in the matrix, say the ith row, comprises the coefficients used in determining the corresponding ith component in the new vector as a linear combination of components of the old vector. In this case the matrix is diagonal, since for each of the four components of the new vector, the only contribution from the old vector comes from the corresponding component. In other

words, the only contribution to $V^{(n)}_{l+\frac{N}{2},m}$ comes from $V^{(o)}_{l+\frac{N}{2},m}$. So, $\vec{V}^{(n)} = \mathbf{L}^h \vec{V}^{(o)}$, where $\vec{V}^{(0)}$ and $\vec{V}^{(n)}$ are the 4x1 vectors whose components have indices (l,m), $(l+\frac{N}{2},m)$, $(l,m+\frac{N}{2})$, and $(l+\frac{N}{2},m+\frac{N}{2})$. The four diagonal entries, $\Lambda_{i,i}$, are:

$$\begin{aligned}
\Lambda_{1,1} &= \left[\frac{h^3}{g384}\left((64j-5)e^{\frac{\imath 2\pi l}{N}} + (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\
&\quad +(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j + (64j+11)e^{\frac{\imath 2\pi m}{N}} \\
&\quad \left.+(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} + (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\
&\quad -\frac{\alpha\kappa h}{2}\left(4je^{\frac{\imath 2\pi l}{N}} + (-1+4j)e^{-\frac{\imath 2\pi m}{N}}\right. \\
&\quad \left.\left.-16j + (1+4j)e^{\frac{\imath 2\pi m}{N}} + 4je^{\frac{-\imath 2\pi l}{N}}\right)\right], \\
\Lambda_{2,2} &= \left[\frac{h^3}{g384}\left(-(64j-5)e^{\frac{\imath 2\pi l}{N}} - (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\
&\quad +(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j + (64j+11)e^{\frac{\imath 2\pi m}{N}} \\
&\quad \left.-(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} - (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\
&\quad -\frac{\alpha\kappa h}{2}\left(-4je^{\frac{\imath 2\pi l}{N}} + (-1+4j)e^{-\frac{\imath 2\pi m}{N}}\right. \\
&\quad \left.\left.-16j + (1+4j)e^{\frac{\imath 2\pi m}{N}} - 4je^{\frac{-\imath 2\pi l}{N}}\right)\right], \\
\Lambda_{3,3} &= \left[\frac{h^3}{g384}\left((64j-5)e^{\frac{\imath 2\pi l}{N}} - (32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right. \\
&\quad -(64j-11)e^{\frac{-\imath 2\pi m}{N}} + 448j - (32j+11)e^{\frac{\imath 2\pi m}{N}} \\
&\quad \left.-(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}} + (64j+5)e^{-\frac{\imath 2\pi l}{N}}\right) \\
&\quad -\frac{\alpha\kappa h}{2}\left(4je^{\frac{\imath 2\pi l}{N}} - (-1+4j)e^{-\frac{\imath 2\pi m}{N}}\right. \\
&\quad \left.\left.-16j - (1+2j)e^{\frac{\imath 2\pi m}{N}} + 4je^{\frac{-\imath 2\pi l}{N}}\right)\right],
\end{aligned}$$

and

$$\begin{aligned}\Lambda_{4,4} = & \left[\frac{h^3}{g384}\left(-(64j-5)e^{\frac{\imath 2\pi l}{N}}+(32j+5)e^{\frac{\imath 2\pi(l+m)}{N}}\right.\right.\\ & -(64j-11)e^{\frac{-\imath 2\pi m}{N}}+448j-(64j+11)e^{\frac{\imath 2\pi m}{N}}\\ & \left.+(32j-5)e^{-\frac{\imath 2\pi(l+m)}{N}}-(64j+5)e^{-\frac{\imath 2\pi l}{N}}\right)\\ & -\frac{\alpha\kappa h}{2}\left(-4je^{\frac{\imath 2\pi l}{N}}-(-1+4j)e^{-\frac{\imath 2\pi m}{N}}\right.\\ & \left.\left.-16j-(1+4j)e^{\frac{\imath 2\pi m}{N}}-4je^{\frac{-\imath 2\pi l}{N}}\right)\right].\end{aligned}$$

A similar process must be applied to determine each of the other matrices that comprise $\mathbf{A}(l,m)$. In order to compute the entries for the matrix $\breve{\mathbf{P}}^h$, recall from Chapter IV, Equation IV.25, that we can write the error equation, $e^{new} = Pe^{old}$, as

$$\begin{aligned}\frac{h^3}{g384}Dv^{(new)}_{2k,2j}-\frac{\alpha\kappa h}{2}Jv^{(new)}_{2k,2j} = & \frac{h^3}{g384}(-Av^{(new)}_{2k-1,2j-1}-Bv^{(new)}_{2k-1,2j}-Cv^{(new)}_{2k,2j-1}\\ & -Ev^{(old)}_{2k,2j+1}-Fv^{(old)}_{2k+1,2j}-Gv^{(old)}_{2k+1,2j+1})\\ & -\frac{\alpha\kappa h}{2}(-Hv^{(new)}_{2k-1,2j}-Iv^{(new)}_{2k,2j-1}\\ & -Kv^{(old)}_{2k,2j+1}-Lv^{(old)}_{2k+1,2j}),\end{aligned}$$

where $A = 32j-5, B = 64j+5, C = 64j-11, D = 448j, E = 64j+11, F = 64j-5, G = 32j+5$, and $H = 4j, I = -1+4j, J = -16j, K = 1+4j, L = 4j$. Using similar calculations to those above, we have the four entries, $\Pi_{i,i}$, of the diagonal matrix $\breve{\mathbf{P}}^h$:

$$\Pi_{1,1} = \frac{\frac{h^3}{g384}[-Ee^{\frac{\imath 2\pi m}{N}}-Fe^{\frac{\imath 2\pi l}{N}}-Ge^{\frac{\imath 2\pi(l+m)}{N}}]-\frac{\alpha\kappa h}{2}[-Ke^{\frac{\imath 2\pi m}{N}}-Le^{\frac{\imath 2\pi l}{N}}]}{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}}+Be^{-\frac{\imath 2\pi l}{N}}+Ce^{\frac{-\imath 2\pi m}{N}}+D]-\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}}+Ie^{-\frac{\imath 2\pi m}{N}}+J]},$$

$$\Pi_{2,2} = \frac{\frac{h^3}{g384}[-Ee^{\frac{\imath 2\pi m}{N}}+Fe^{\frac{\imath 2\pi l}{N}}+Ge^{\frac{\imath 2\pi(l+m)}{N}}]-\frac{\alpha\kappa h}{2}[-Ke^{\frac{\imath 2\pi m}{N}}+Le^{\frac{\imath 2\pi l}{N}}]}{\frac{h^3}{g384}[-Ae^{-\frac{\imath 2\pi(l+m)}{N}}-Be^{-\frac{\imath 2\pi l}{N}}+Ce^{\frac{-\imath 2\pi m}{N}}+D]-\frac{\alpha\kappa h}{2}[-He^{\frac{-\imath 2\pi l}{N}}+Ie^{-\frac{\imath 2\pi m}{N}}+J]},$$

$$\Pi_{3,3} = \frac{\frac{h^3}{g384}[Ee^{\frac{\imath 2\pi m}{N}}-Fe^{\frac{\imath 2\pi l}{N}}+Ge^{\frac{\imath 2\pi(l+m)}{N}}]-\frac{\alpha\kappa h}{2}[Ke^{\frac{\imath 2\pi m}{N}}-Le^{\frac{\imath 2\pi l}{N}}]}{\frac{h^3}{g384}[-Ae^{-\frac{\imath 2\pi(l+m)}{N}}+Be^{-\frac{\imath 2\pi l}{N}}-Ce^{\frac{-\imath 2\pi m}{N}}+D]-\frac{\alpha\kappa h}{2}[He^{\frac{-\imath 2\pi l}{N}}-Ie^{-\frac{\imath 2\pi m}{N}}+J]},$$

and

$$\Pi_{4,4} = \frac{\frac{h^3}{g384}Ee^{\frac{\imath 2\pi m}{N}}+Fe^{\frac{\imath 2\pi l}{N}}-Ge^{\frac{\imath 2\pi(l+m)}{N}}]-\frac{\alpha\kappa h}{2}[Ke^{\frac{\imath 2\pi m}{N}}+Le^{\frac{\imath 2\pi l}{N}}]}{\frac{h^3}{g384}[Ae^{-\frac{\imath 2\pi(l+m)}{N}}-Be^{-\frac{\imath 2\pi l}{N}}-Ce^{\frac{-\imath 2\pi m}{N}}+D]-\frac{\alpha\kappa h}{2}[-He^{\frac{-\imath 2\pi l}{N}}-Ie^{-\frac{\imath 2\pi m}{N}}+J]}.$$

In similar fashion, the entry for the 1x1 matrix $(\check{\mathbf{L}}^H)^{-1}$, corresponding to the coarse grid relaxation operator, is given by

$$\frac{\frac{h^3}{g384}[-E^H e^{\frac{\imath 2\pi m}{N}} - F^H e^{\frac{\imath 2\pi l}{N}} - G^H e^{\frac{\imath 2\pi(l+m)}{N}}] - \frac{\alpha\kappa h}{2}[-K^H e^{\frac{\imath 2\pi m}{N}} - L^H e^{\frac{\imath 2\pi l}{N}}]}{\frac{h^3}{g384}[A^H e^{-\frac{\imath 2\pi(l+m)}{N}} + B^H e^{-\frac{\imath 2\pi l}{N}} + C^H e^{\frac{-\imath 2\pi m}{N}} + D^H] - \frac{\alpha\kappa h}{2}[H^H e^{\frac{-\imath 2\pi l}{N}} + I^H e^{-\frac{\imath 2\pi m}{N}} + J^H]};$$

where $A^H = 16j-5, B^H = 32j+5, C^H = 32j-11, D^H = 224j, E^H = 32j+11, F^H =$

$32j-5, G^H = 16j+5$, and $H^H = 2j, I^H = -1+2j, J^H = -8j, K^H = 1+2j, L^H = 2j$.

The equation for the conservation restriction operator, I_h^H, is

$$\begin{aligned} v_{k,j}^H &= \frac{(2j-\frac{3}{4})}{4(2j-1)}\left\{v_{2k-1,2j-1}^h + 2v_{2k,2j-1}^h + v_{2k+1,2j-1}^h\right\} \\ &+\frac{1}{2}\left\{v_{2k-1,2j}^h + 2v_{2k,2j}^h + v_{2k+1,2j}^h\right\} \\ &+\frac{(2j+\frac{3}{4})}{4(2j+1)}\left\{v_{2k-1,2j+1}^h + 2v_{2k,2j+1}^h + v_{2k+1,2j+1}^h\right\}. \end{aligned}$$

Let $Q = \frac{(2j-\frac{3}{4})}{(2j-1)}$ and $R = \frac{(2j+\frac{3}{4})}{(2j+1)}$ and expand this relation in a DFT. Considering those frequencies that can be represented on the coarse grid and, making use of the orthogonality property, we can equate individual terms of the sum. We divide by $\hat{C}(l,m) = e^{\frac{\imath 2\pi kl}{N}} e^{\frac{\imath 2\pi jm}{N}}$, the coarse grid analog to $C(l,m)$, to give the components of the matrix of coefficients corresponding to the restriction operator, $\check{\mathbf{I}}_h^H = (\Delta_{1,i})$. This matrix is 1x4 since the output of the operator, a single component on the coarse grid, is a linear combination of 2^2 components on the fine grid. The entries in the matrix are as follows:

$$\begin{aligned} \Delta_{1,1} &= \frac{1+cos(\frac{2\pi l}{N})}{2}\left[Qe^{-\frac{\imath 2\pi m}{N}} + 2 + Re^{\frac{\imath 2\pi m}{N}}\right], \\ \Delta_{1,2} &= \frac{1-cos(\frac{2\pi l}{N})}{2}\left[Qe^{-\frac{\imath 2\pi m}{N}} + 2 + Re^{\frac{\imath 2\pi m}{N}}\right], \\ \Delta_{1,3} &= \frac{1+cos(\frac{2\pi l}{N})}{2}\left[-Qe^{-\frac{\imath 2\pi m}{N}} + 2 - Re^{\frac{\imath 2\pi m}{N}}\right], \quad \text{and} \\ \Delta_{1,4} &= \frac{1-cos(\frac{2\pi l}{N})}{2}\left[-Qe^{-\frac{\imath 2\pi m}{N}} + 2 - Re^{\frac{\imath 2\pi m}{N}}\right]. \end{aligned}$$

The equation for the injection restriction operator is $v^H_{k,j} = v^h_{2k,2j}$ which, using similar calculation, results in

$$V^H_{l,m} = V^h_{l,m} + V^h_{l+\frac{N}{2},m} + V^h_{l,m+\frac{N}{2}} + V^h_{l+\frac{N}{2},m+\frac{N}{2}},$$

so that $\breve{\mathbf{I}}^H_h = [1 \quad 1 \quad 1 \quad 1]$.

The linear interpolation operator gives rise to the following relations:

$$\begin{aligned}
v^h_{2k,2j} &= v^H_{k,j}, \\
v^h_{2k+1,2j} &= \frac{v^H_{k,j} + v^H_{k+1,j}}{2}, \\
v^h_{2k,2j+1} &= \frac{v^H_{k,j} + v^H_{k,j+1}}{2}, \\
v^h_{2k+1,2j+1} &= \frac{v^H_{k,j} + v^H_{k+1,j+1}}{2}.
\end{aligned}$$

Expanding these in a DFT, considering the coarse grid frequencies, and making use of the orthogonality property, we can equate terms of the sums and divide by $\hat{C}(l,m)$ to give

$$\begin{aligned}
V^H_{l,m} &= V^h_{l,m} + V^h_{l+\frac{N}{2},m} + V^h_{l,m+\frac{N}{2}} + V^h_{l+\frac{N}{2},m+\frac{N}{2}}, \\
V^H_{l,m}\frac{(e^{\frac{\imath 2\pi l}{N}} + e^{\frac{-\imath 2\pi l}{N}})}{2} &= V^h_{l,m} - V^h_{l+\frac{N}{2},m} + V^h_{l,m+\frac{N}{2}} - V^h_{l+\frac{N}{2},m+\frac{N}{2}}, \\
V^H_{l,m}\frac{(e^{\frac{\imath 2\pi m}{N}} + e^{\frac{-\imath 2\pi m}{N}})}{2} &= V^h_{l,m} + V^h_{l+\frac{N}{2},m} + V^h_{l,m+\frac{N}{2}} + V^h_{l+\frac{N}{2},m+\frac{N}{2}}, \quad \text{and} \\
V^H_{l,m}\frac{(e^{\frac{\imath 2\pi l+m}{N}} + e^{\frac{-\imath 2\pi (l+m)}{N}})}{2} &= V^h_{l,m} + V^h_{l+\frac{N}{2},m} + V^h_{l,m+\frac{N}{2}} + V^h_{l+\frac{N}{2},m+\frac{N}{2}}.
\end{aligned}$$

The components, $\Theta_{i,1}$, of the matrix of coefficients, $\breve{\mathbf{I}}^h_H$, are found by solving the above system of four equations for the fine grid components, such that $\vec{V}^h = \breve{\mathbf{I}}^h_H \vec{V}^H$. The matrix $\breve{\mathbf{I}}^h_H$ is 4x1 since the output of the operator, the 2^2 components of the fine grid vector, is the result of interpolation operator acting on the single component of the

coarse grid vector. That is,

$$\begin{bmatrix} \Theta_{1,1} \\ \Theta_{2,1} \\ \Theta_{3,1} \\ \Theta_{4,1} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 + cos(\frac{2\pi l}{N}) + cos(\frac{2\pi m}{N}) + cos(\frac{2\pi(l+m)}{N}) \\ 1 - cos(\frac{2\pi l}{N})cos(\frac{2\pi m}{N}) - cos(\frac{2\pi(l+m)}{N}) \\ 1 + cos(\frac{2\pi l}{N}) - cos(\frac{2\pi m}{N}) + cos(\frac{2\pi(l+m)}{N}) \\ 1 - cos(\frac{2\pi l}{N}) - cos(\frac{2\pi m}{N}) + cos(\frac{2\pi(l+m)}{N}) \end{bmatrix}.$$

Thus, we have all the elements to construct the amplification matrix, $\mathbf{A}(l, m)$, (Equation VI.3). We now have the tools to analyze the expected perfomance of a two-level scheme.

C. NUMERICAL RESULTS

We now conduct a number of numerical experiments to investigate the performance of the multigrid method, again using *Matlab*. First we examine the behavior of the amplification matrix, $\mathbf{A}(l, m)$, and the largest spectral radius of the family of matrices $\mathbf{A}(l, m)$ for all l, m. We then present the results of the repeated use of the V-cycle to solve Equation IV.2, as compared with the iterative solutions.

In order to make use of the amplification matrix, we must choose a relaxation method, and the number of times to relax on the way down and on the way back up, i.e., we must choose ν_1 and ν_2 in Equation VI.3. We experiment first with Gauss-Seidel relaxation and three different combinations of (ν_1, ν_2) V-cycles: a $(0, 1)$ cycle, a $(2, 1)$ cycle, and a $(10, 10)$ cycle (recall that $\alpha = 1$, meaning fully implicit time stepping, and the time step $g = h$). The first two types of cycles are fairly common (see [Ref. 3]), while the latter is included for reference. The information presented in Tables VIII, IX, and X indicates the asymptotic convergence factor, $\bar{\lambda}$, for specific grid sizes, grid positions, and type of restriction operator, either conservation or injection. Linear interpolation is used as the interpolation operator in all of these experiments. Tables XI and XII indicate the asymptotic convergence factors for horizontal and vertical line relaxation and a $(2, 1)$ cycle.

Table VIII indicates the asymptotic convergence factors that result from a

$(0,1)$ cycle, Gauss-Seidel						
	Conservation			Injection		
$2j$	$N=16$	$N=32$	$N=64$	$N=16$	$N=32$	$N=64$
2	.9174	.9651	.9843	.9397	.9710	.9858
$\frac{N}{2}$	.8308	.9111	.9544	.9144	.9553	.9771
$N-2$	.7472	.8595	.9259	.8929	.9422	.9700

Table VIII. Asymptotic Convergence Factors for the $(0,1)$ Cycle Using Gauss-Seidel Relaxation.

single iteration sweep on the fine grid coupled with the course grid correction for both the conservation and injection restriction operators. In general, the conservation restriction operator seems to indicate better performance, since its convergence factors are lower than those for the injection operator (recall that the asymptotic convergence factor is the largest spectral radius of the family of amplification matrices, which comes from the DFT expansion of each component in the two-level error propagation matrix). Similar to the analysis on relaxation schemes in Chapter IV, the convergence factors decrease with radial distance, indicating an improvement in performance, but increase with grid size. Even though the convergence factors for all three grid sizes are less than one, indicating that convergence is very likely (but not necessarily certain, since these factors tell us nothing about what is going on at the boundaries), at least some of the factors for all three grids are relatively high, indicating that convergence may be quite slow with the $(0,1)$ cycle. A point of interest is to compare these results with those in Table IV, which indicate the maximum ratio of new to old error components for Gauss-Seidel relaxation. One might expect the results in Table VIII to be universally better than those in Table IV, however this is not the case. In fact, on a grid of size $N = 32$, the largest asymptotic convergence factors for both the conservation and the injection restriction operators are larger than the largest ratio for just a single Gauss-Seidel sweep on the fine grid. In other words, a single Gauss-Seidel relaxation sweep is predicted to do better than a $(0,1)$ cycle. This is a

(2, 1) cycle, Gauss-Seidel						
	Conservation			Injection		
$2j$	$N = 16$	$N = 32$	$N = 64$	$N = 16$	$N = 32$	$N = 64$
2	.8223	.9133	.9574	.8423	.9189	.9589
$\frac{N}{2}$	.7375	.8572	.9254	.8116	.8987	.9474
$N - 2$	.6623	.8083	.8977	.7914	.8861	.9404

Table IX. Asymptotic Convergence Factors for the (2, 1) Cycle Using Gauss-Seidel Relaxation.

fact worth remembering when the multilevel solution is analyzed.

Table IX indicates the asymptotic convergence factors for a (2, 1) cycle; as expected, the factors are lower than the corresonding factors for the (0, 1) cycle, and are universally lower than the ratios associated with a single Gauss-Seidel sweep as indicated in Table IV. However, another comparison is to raise the ratios in Table IV to the third power, corresponding to three relaxation sweeps on the fine grid. In other words, since for a (2, 1) cycle there are two relaxation sweeps on the fine grid before the course grid correction, and one afterward, for a total of three relaxation sweeps on the fine grid, it seems reasonable to compare the performance of a (2, 1) cycle to three iterations of the Gauss-Seidel method, represented by raising the Gauss-Seidel factor to the third power. For example, in Table IV for $N = 32$, $j = 1$, the ratio is .9567 which, when raised to the third power is .8756. This value is lower than either of the values in Table IX for $N = 32$, $2j = 2$. This also holds true for the asymptotic convergence factor associated with $N = 32$ and $2j = \frac{N}{2}$ (this phenomenon does not occur for $N = 32$, $2j = N - 2$, nor for $N = 16$). Thus, the overall predicted performance of three Gauss-Seidel relaxation sweeps for $N = 32$ is better than the predicted performance of a Gauss-Seidel (2, 1) cycle.

Similar to the information in Table VIII, the performance predictions in Table IX improve with radial distance, and worsen with increase in grid size. Also, as

(10, 10) cycle, Gauss-Seidel						
	Conservation			Injection		
$2j$	$N = 16$	$N = 32$	$N = 64$	$N = 16$	$N = 32$	$N = 64$
2	.3243	.5716	.7567	.3322	.5751	.7579
$\frac{N}{2}$	.2678	.5104	.7118	.2948	.5351	.7288
$N - 2$	.2374	.4795	.6898	.2837	.5256	.7226

Table X. Asymptotic Convergence Factors for the (10, 10) Cycle Using Gauss-Seidel Relaxation.

above, the conservation restriction appears to predict better performance than injection. The factors are still relatively high, especially for the $N = 64$ grid size, so that convergence will likely be slow.

The (10, 10) cycle is included for reference so that the performance of the two-level cycle can analyzed using a large number of iteration sweeps on the fine grid. The characteristics of the (10, 10) cycle are similiar to those of the (0, 1) cycle and the (2, 1) cycle; the performance improves with radial distance and worsens with increase in grid size. The asymptotic convergence factors are smaller than for previous cycles, but this is expected since a good deal more work is represented by the (10, 10) cycle. However, using a comparison similar to that in the discussion of Table IX, in Table IV for $N = 32$, $j = 1$, the ratio is .9567 which, when raised to the 20th power is .4126. Here, as in the case of a (2, 1) cycle, this value is lower than either of the values in Table X for $N = 32$, $j = 1$. This holds true for all the factors associated with $N = 32$, but is not true for $N = 16$. Thus, as before, the predicted performance of Gauss-Seidel relaxation for $N = 32$ is better than the predicted performance of a Gauss-Seidel two-level cycle.

Tables XI and XII indicate the asymptotic convergence factors for horizontal and vertical line relaxation (2, 1) cycles. Similar to the results of the analysis on the horizontal and line relaxation schemes in Chapter IV, both line relaxation schemes for the (2, 1) cycle indicate better performance than that for the Gauss-Seidel method;

(2, 1) cycle, Horizontal Line Relaxation						
	Conservation			Injection		
$2j$	$N = 16$	$N = 32$	$N = 64$	$N = 16$	$N = 32$	$N = 64$
2	.6784	.8262	.9097	.6938	.8310	.9111
$\frac{N}{2}$	.6210	.7836	.8839	.6793	.8203	.9046
$N - 2$	.5629	.7408	.8580	.6647	.8096	.8981

Table XI. Asymptotic Convergence Factors for the (2, 1) Cycle Using Horizontal Line Relaxation.

(2, 1) cycle, Vertical						
	Conservation			Injection		
$2j$	$N = 16$	$N = 32$	$N = 64$	$N = 16$	$N = 32$	$N = 64$
2	.7258	.8562	.9265	.7429	.8613	.9279
$\frac{N}{2}$	.6326	.7876	.8851	.6928	.8247	.9059
$N - 2$	.5685	.7427	.8586	.6723	.8119	.8988

Table XII. Asymptotic Convergence Factors for the (2, 1) Cycle Using Vertical Line Relaxation.

horizontal line relaxation is the better of the two. One difference is that in Tables V and VI, the predictions for horizontal line relaxation worsen, and the predictions for vertical line relaxation improve, with radial distance, whereas in Tables XI and XII the predictions for both line relaxation schemes improve with radial distance. Using calculations similar to those in the discussions of Tables IX and X, the values in Tables V and VI are raised to the third power and compared with the factors in Tables XI and XII. The result is that the predicted performance of three sweeps of the horizontal line relaxation is better than that for the (2, 1) cycle for both $N = 16$ and $N = 32$. The predicted performance of three sweeps of the vertical line relaxation is better than that for the (2, 1) cycle only for $N = 32$. These results are consistent with those of the Gauss-Seidel relaxation for all of the (ν_1, ν_2) cycles tested. That is, for $N = 32$, the predicted performance of ordinary iteration is better than the predictions for any of the corresponding two-level schemes considered here.

In the initial stages of using V-cycles to solve Equation IV.2 at a single time step, we experiment with the use of conservation restriction and linear interpolation. The result is that this combination does not result in a coarse grid correction. That is, the effect of applying the two-level scheme is to generate an approximation that is not as good as that obtained by relaxation on the fine grid alone. Therefore, we experiment with the combination of the injection restriction operator and the linear interpolation operator, with the result that this combination does produce a coarse grid correction. A correction scheme using V-cycles with these intergrid transfer operators is implemented to solve at a single time step, and then the solution is stepped in time until a "steady" solution was obtained.

Multigrid V-cycle solutions to Equation IV.2 are computed on grids $N = 8, 16,$ and 32, using (2, 1) cycles, Gauss-Seidel relaxation, $\alpha = 1$ (fully implicit time stepping), and time step $g = h$, where the initial temperature distribution and boundary conditions are the same as in the iterative solution. We again use the discrete energy norm (Equation V.1) to measure the accuracy of these solutions,

$$|||D^h||| = \langle L^h D^h, D^h \rangle^{\frac{1}{2}},$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, L^h is the operator defined in Equation IV.2, and $D^h = T^h - T^*$ approximates the discretization error, where T^h is the solution to Equation IV.2 on grid h, and T^* is the "exact" solution (i.e., solution of Equation IV.2 with $N = 64$ sampled on grid h) (see [Ref. 11]). The results presented in Table XIII are almost identical to those obtained for the iterative solution (see Chapter V).

As another measure of how well the iterative solutions match the multilevel solutions, the discrete L^2 norm of the solution differences, d^h,

$$\|d^h\| = \frac{1}{N} \langle d^h, d^h \rangle^{\frac{1}{2}},$$

is computed for each of the above grid sizes and time value. The results are presented in Table XIV; while the norm of the differences is generally small, it increases with

	$N = 8$	$N = 16$	$N = 32$
Initial time step	43.07	23.97	6.87
Time = "1 second"	42.99	23.98	6.87
Steady state	42.99	23.98	6.87

Table XIII. Discretization Errors for the Multilevel Solution.

	$N = 8$	$N = 16$	$N = 32$
Initial time step	8.00E-16	1.74E-15	3.95E-15
Time = "1 second"	5.68E-16	1.53E-15	4.64E-15
Steady state	3.02E-15	1.02E-14	5.66E-14

Table XIV. Norms of Differences Between Iterative and Multilevel Solutions.

grid size. This might suggest a normalization problem, however, the factor of $\frac{1}{N}$ in the discrete L^2 norm is used specifically to avoid this type of problem. More investigation is required to determine why there is not better agreement between the iterative and multilevel solutions.

Figures 30 and 31 indicate the number of iterations per time step for the iterative solutions and the number of V-cycles per time step for the multigrid solution for $N = 32$. It is important to note that, while both processes require a decreasing amount of work for each time step out to about 150 steps, the multilevel process requires about $\frac{1}{3}$ more time steps than the iterative process to reach a steady solution: the multilevel process requires one V-cycle per time step for time step > 150. In addition, Figures 32 and 33 illustrate how the norm of the difference between successive solutions behaves as the solution is stepped in time. As would be hoped, the difference between successive solutions decreases with time stepping. However, as is evident from these latter figures, the time stepping process stalls somewhat before the steady solution is reached (the tolerance used is machine $\epsilon = 2.22E - 16$). While the number of time steps to reach a steady solution on grids $N = 8, 16$ (not shown) is about the same for both the iterative and multilevel solutions, on grid $N = 32$,

as already indicated, the number of time steps needed for the multilevel process to reach steady solution is about $\frac{1}{3}$ longer than that required for the iterative solution. Even so, the two processes on all three grid sizes stall at about the same time step.

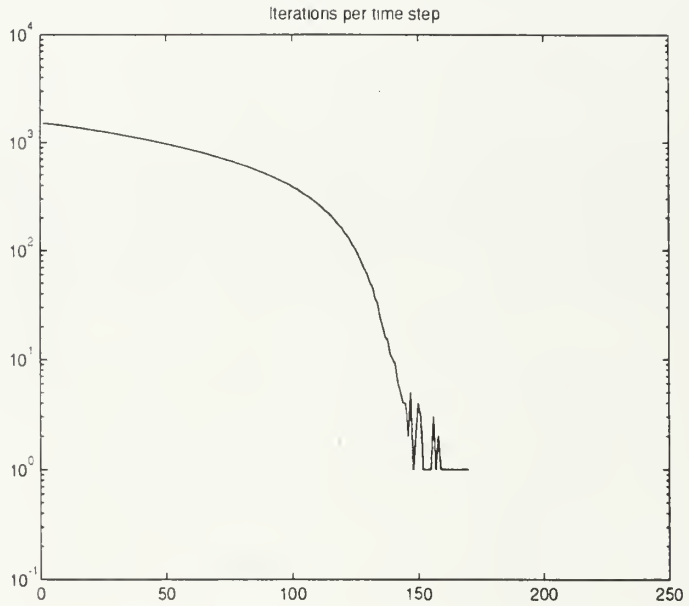

Figure 30. The number of iterations per time step for $N = 32$; 170 time steps needed to reach a steady solution.

We conclude our analysis of the multilevel solution with a discussion of how much computational work must be done to reach a solution. For this purpose, we pick a time step, for both methods, for which the time stepping process has stalled. For $N = 8$, use time step $s = 50$; for $N = 16$, use time step $s = 80$; and for $N = 32$, use a time step of $s = 150$. The norm of the difference between successive solutions for these time steps is on the order of 10^{-15}. We use calculations similar to those in [Ref. 3] in computing the cost. If we consider a **work unit, WU**, to be the cost of one relaxation sweep on the finest grid, then we can estimate how many work units are associated with each V-cycle. Since we are using a $(2, 1)$ cycle, relaxation occurs three times on each level; one relaxation sweep on the grid Ω^{ph} requires p^{-d} of a work unit, where d is the dimension. Adding these costs, and using the geometric series as

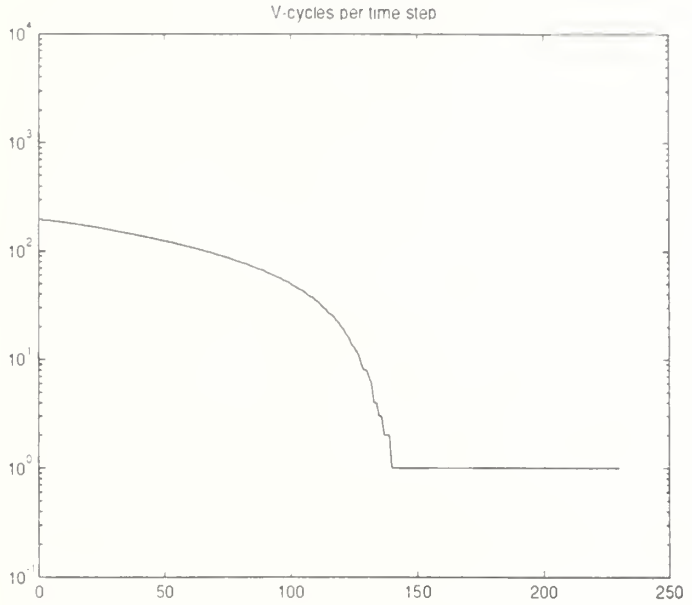

Figure 31. The number of V-cycles per time step for $N = 32$; 230 time steps needed to reach a steady solution.

an upper bound, we have

$$\text{V-cycle Cost} = 3\{1 + 2^{-d} + \cdots + 2^{-nd}\}\text{WU} < \frac{3}{1 - 2^{-d}}\text{WU}.$$

Thus, for the two-dimensional case we have Cost = 4WU. A comparison of the computational cost for the various grid sizes is given in Table XV.

	$N = 8$	$N = 16$	$N = 32$
Time step (s)	50	80	150
"Elapsed" time ($\frac{s}{N}$)	6.25	5.00	4.69
Number of V-cycles	1246	4076	13281
Total Cost (WU)			
V-cycle cost (WU)	4984	16304	53124
Iteration cost (WU)	6245	25631	103185

Table XV. Computational Cost for the Iterative and Multigrid Solutions (Not Including the Cost of the Intergrid Transfers).

Figure 32. Norm of difference between successive iterative solutions for $N = 32$; 170 time steps needed to reach a steady solution.

The cost of implementing the intergrid transfers is not included in the computations presented in Table XV. In considering the amount of work represented by one work unit, counting "adds" and "multiplies" each as one floating point operation (flop), the number of flops for one relaxation sweep on a grid of size N is approximately $41N^2$. Injection restriction requires no arithmetic; linear interpolation requires approximately $\frac{3N^2}{8}$ flops. Thus, the cost of the intergrid transfers for the two-level scheme is about $(\frac{3N^2}{8}/41N^2)WU = \frac{3}{328}WU$. Again using the geometric series as an upper bound, we have that the cost of the intergrid transfers is given by

$$\text{Transfer Cost} = \frac{3}{328}\{1 + 2^{-d} + \cdots + 2^{-nd}\}\text{WU} < \frac{3}{246}\text{WU}.$$

Thus, we update the information in Table XV and present the results in Table XVI.

While not dramatic, the use of multigrid does result in a savings of computational cost. The cost savings increase with grid size; the savings on a grid of size $N = 8$ is only 20%, on a grid of size $N = 32$ the savings is almost 50%. In considering these results, several questions need to be answered. For example, are these

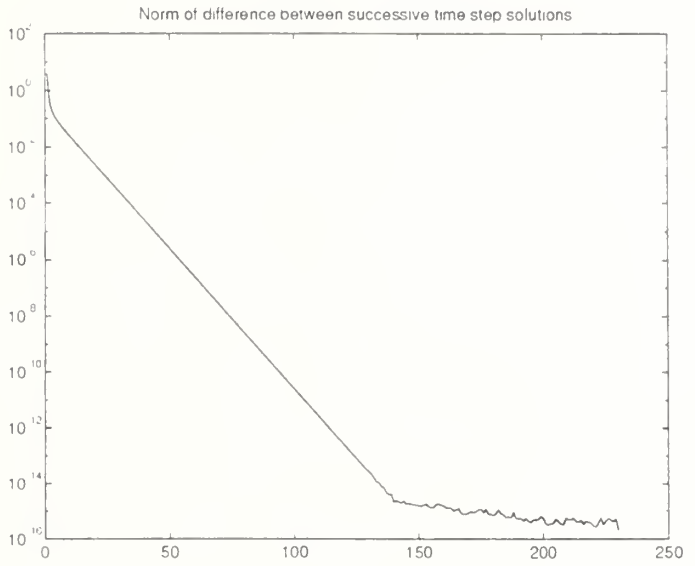

Figure 33. Norm of difference between successive multilevel solutions for $N = 32$; 230 time steps needed to reach a steady solution.

	$N = 8$	$N = 16$	$N = 32$
Total Cost (WU)			
V-cycle cost (WU)	4992	16354	53286
Iteration cost (WU)	6245	25631	103185

Table XVI. Computational Cost for the Iterative and Multigrid Solutions, Including the Cost of the Integrid Transfers.

results consistent with the local mode analysis and the information that comes from the amplification matrices? Does the multigrid solution show "typical" performance; does the process work as one would hope?

In general, the results are consistent with the local mode analysis, which predicts that convergence will be slow. The prediction of the amplification matrices is difficult to apply, since the asymptotic convergence factors apply to two-level schemes and not to V-cycles. Nevertheless, the convergence factors did not predict that the two-level scheme would provide improvement over the iterative process on

grid size $N = 32$, and therefore these results are somewhat of a pleasant surprise. In other words, multigrid performed better than might have been expected based on the two-level convergence factors. However, typical multigrid performance reaches convergence (to the level of truncation error, see [Ref. 3] and [Ref. 2]) in just a few V-cycles. Thus, the multilevel solution developed here is disappointing, since convergence requires a very large number of V-cycles.

There are no obvious causes for the disappointing performance of the multilevel solution of Equation II.2. One of the problems may be the use of cylindrical coordinates, which results in a radial bias in the FVE stencils. The discretization of the time derivative using finite differences, while using the FVE method discretize the spatial derivatives, may also have some impact on the solution process. It is not clear, however, that either of these affects the multilevel solution any differently than it affects the iterative solution. The choice of relaxation scheme is a major factor in determining the convergence of the process; perhaps the use of line relaxation, or some other type smoother, will improve the performance. Again, however, it is not certain that this will result in an improvement of the multigrid performance relative to the iterative process. The determination of the coarse grid and intergrid transfer operators is also something of a concern. This is one area that clearly affects the performance of the multilevel solution as compared to the iterative solution. The fact that the predicted performance of iteration sweeps alone is better than the predicted performance of the two-level schemes emphasizes the need to consider the use of different coarse grid and intergrid transfer operators. Finally, the point-source problem includes a singularity at the boundary. Most of the analysis has been done for interior points, and the solution results do not drastically differ from what is predicted by the analysis. Neverthless, the treatment of the boundaries, particularly for this problem, may provide a means to improve the solution process.

VII. CONCLUSION

The purpose of this work is to apply the FVE method to discretize the axisymmetric heat equation, and implement multigrid in solving it. In addition to using the FVE method to discretize the spatial derivatives, finite differences are applied to the time derivatives, resulting in a time-stepping scheme that makes use of a weighted average of the current and future time steps. The use of cylindrical coordinates results in a discretization stencil that has a radial bias, the effect of which is evident in the analysis of various relaxation schemes. Gauss-Seidel is the relaxation scheme chosen for implementation in the iterative and multilevel solution processes. Due to numerical anomalies encountered in computing the iterative solution, the weighted-average time-stepping scheme becomes a fully implicit backward time-stepping scheme. Its use requires more computational work than the weighted-average scheme, but its use is required to achieve a physically meaningful solution. The specifics of the multilevel technique are then presented, including the development of the coarse grid and intergrid transfer operators, and the predictions of expected performance for the two-level scheme. The multilevel solution is then implemented; its results are analyzed and compared to the results of the iterative solution process. The results of our work are somewhat disappointing in that we are not yet able to achieve the hoped-for level of success. There are, however, a number of positive results that come from this work, specifically, the application of the FVE method to a problem in cylindrical coordinates and the use of *Maple* software to compute the necessary integrals. Additionally, we know that the usual multilevel techniques do not produce the expected results and therefore there is ample room for further study.

There are several possibilities for future work. One specific question that has not been addressed is the determination of the consistency of the discretization operator. Numerical results are useful in attempting to verify this result, but consistency

should be proven analytically. In conjunction with this idea, there may be a requirement to review the assumption that the solution can be adequately represented by a collection of continuous, piecewise linear basis functions. There is no evidence as yet that would require this review, but it is possible. Additionally, we are assuming a uniform step size that applies to both the r and z directions. One possible modification to this approach is to implement a non-uniform grid, perhaps one that results in control volumes that are the same for every point on the grid. The point of this modification would be to eliminate the radial bias in the discretization. For the discretization of the time derivatives, the requirement to produce a physically meaningful solution has resulted in a fully implicit time-stepping scheme. While there may be another method to discretize the time derivatives, it will have to meet the same criterion and, therefore, may not be much of an improvement.

All of the above considerations apply generally to the problem, and not specifically to multigrid. There are at least three areas in which multigrid performance may be improved: intergrid transfer operators, coarse grid operator, and treatment of boundary conditions. The intergrid transfer operators used are the simplest available that result in coarse grid correction and, consequently, choosing different operators may improve performance. Other possibilities include the use of a restriction operator that is based on the piecewise linearity of the basis functions, or perhaps an interpolation operator that is based on enforcing conservation. In other words, choose intergrid operators that satisfy the *variational condition*, $I_H^h = c(I_h^H)^T$, for c some constant (see [Ref. 3] and [Ref. 1]). The choice of the coarse grid operator may need to be made in conjunction with these intergrid operators. That is, despite the computational complexities that would be introduced, choose a coarse grid operator such that the *Galerkin condition* $L^h = I_H^h L^H I_h^H$ (see [Ref. 3] and [Ref. 1]) is satisfied. Additionally, special treatment of the boundaries may be required, since the boundary of the point-source problem has a singularity at the origin (see [Ref. 2]).

Finally, the consideration of the point-source problem is the first step in ap-

plying the FVE discretization and multilevel solution to the Navier-Stokes equation that arises from the molten-pool problem. The intricacies of the molten-pool problem present several challenges. Included are the requirement to solve a system of three PDEs in three unknowns and the requirement to keep track of the moving phase-change boundary as the molten pool expands. Additionally, high flow velocities and small local length scales result when convection is vigorous ([Ref. 13]), causing further numerical complications. It is anticipated that the Full Approximation Scheme (FAS) (see [Ref. 2] and [Ref. 1]) can be used to effectively treat the non-linear nature of the problem, and that the Fast Adaptive Composite Grid (FAC) method (see [Ref. 1] will be useful in overcoming difficulties that arise in the geometry of the problem.

APPENDIX A. THE DISCRETE FOURIER TRANSFORM

Below we briefly list the definition and some of the properties of the **Discrete Fourier Transform, DFT.** For an exhaustive treatment see [Ref. 10]; what appears in this appendix closely follows that treatment.

The DFT is, in at least one interpretation, a way to discretely approximate the Fourier transform. For example, assume we are working on a temporal or spatial domain $[-\frac{A}{2}, \frac{A}{2}]$ with grid spacing $\Delta x = A/N$ and grid points $x_j = j\Delta x$, and its associated frequency domain $[-\frac{\Omega}{2}, \frac{\Omega}{2}]$ with grid spacing $\Delta\omega = 2\pi/A$ and grid points $\omega_m = m\Delta\omega$, where $j, m = -\frac{N}{2}+1 : \frac{N}{2}$. Suppose v_j denotes the sampled values, $v(x_j)$, of a function defined on the domain $[-\frac{A}{2}, \frac{A}{2}]$, and that $\hat{v}(\omega_m)$ is the Fourier transform of v at the frequency grid points, ω_m, where the Fourier transform is given by

$$\hat{v}(\omega) \equiv \int_{-\infty}^{\infty} v(x) e^{-i2\pi\omega x} dx.$$

Using the Trapezoidal Rule to approximate the integral, we have

$$\hat{v}(\omega_m) = \int_{-\frac{A}{2}}^{\frac{A}{2}} v(x) e^{-\frac{i2\pi m x}{A}} dx \approx A \left\{ \frac{1}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} v_j e^{\frac{-i2\pi jm}{N}} \right\},$$

where $m = -\frac{N}{2} + 1 : \frac{N}{2}$. Given the set of N sample values of v_j, the DFT comprises the N coefficients

$$V_m = \frac{1}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} v_j e^{\frac{-i2\pi jm}{N}}, \quad \text{for} \quad \mathrm{m} = -\frac{\mathrm{N}}{2} + 1 : \frac{\mathrm{N}}{2}.$$

Thus, the DFT can be considered to approximate the Fourier transform $\hat{v}(\omega_m)$ by $\hat{v}(\omega_m) \approx AV_m$.

We now give a formal definition of the DFT[1].

---

[1]There are several definitions of the DFT, as indicated in [Ref. 10].

Definition A.1 *Let N be an even positive integer and $\{v_j\}$ a sequence[2] of N complex numbers where $j = -\frac{N}{2}+1 : \frac{N}{2}$. The discrete Fourier transform of the sequence $\{v_j\}$ is another sequence of N complex numbers, $\{V_m\}$, whose elements are given by*

$$V_m = \frac{1}{N} \sum_{j=-\frac{N}{2}+1}^{\frac{N}{2}} v_j e^{\frac{-\imath 2\pi jm}{N}}, \tag{A.1}$$

for $m = -\frac{N}{2}+1 : \frac{N}{2}$.

The DFT may be defined for N odd as well, however we will not need this definition for our discussion. The choice of indices, $-\frac{N}{2}+1 : \frac{N}{2}$ as opposed to $0 : N-1$, is made deliberately in order to simplify some of the analysis that is to follow from applying the DFT, and because it is more natural, if less familiar. While the choice of indices does not affect the applicability of the transform, care must be taken to avoid confusion over which grid points correspond to what type of frequency (see Figure 34). We use the operator notation $\mathcal{D}\{v_j\}$ to denote the DFT of the sequence $\{v_j\}$, and $\mathcal{D}\{v_j\}_m$ to indicate the mth element of the transform, $\mathcal{D}\{v_j\}_m = V_m$.

In general, the output of the DFT, $\{V_m\}$, is a complex-valued sequence. The interpretation of the DFT coefficients is essentially the same as that given for the (continuous) Fourier transform. Also, V_m is even more closely related to c_m, the Fourier series coefficient. In many ways, the DFT is more naturally viewed as an approximation to c_m than to $\hat{v}(\omega)$. The mth DFT coefficient V_m gives the "amount" of the mth mode (with a frequency ω_m) that is present in the input sequence v_j. In contrast to the use of modes of all frequencies, as in the Fourier transform, an N-point DFT uses only N distinct modes, with roughly $\frac{N}{2}$ different frequencies. The modes can be labeled by the frequency index m, and each mode has a value at each grid point x_j where $j = -\frac{N}{2}+1 : \frac{N}{2}$. Therefore we denote the jth component of the mth DFT mode as

$$\omega_N^{-jm} = e^{\frac{-\imath 2\pi jm}{N}},$$

for $j, m = -\frac{N}{2}+1 : \frac{N}{2}$.

---

[2]The terms *sequence* and *vector* are used interchangeably to denote the input/output of the DFT.

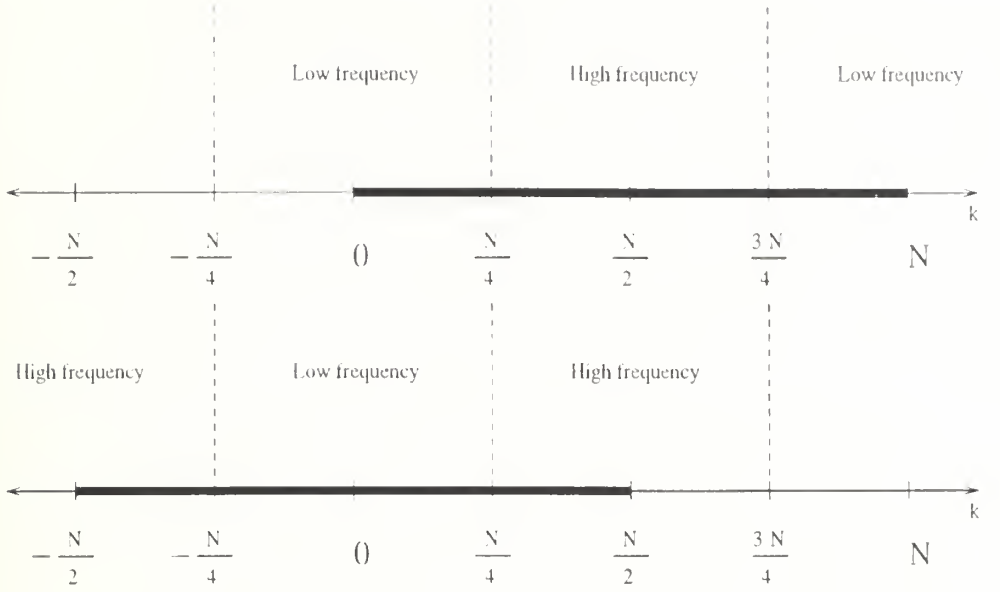

Figure 34. Using centered indices (bottom figure), the low frequencies correspond to $|k| < \frac{N}{4}$ for a single set of sample points; the high frequencies correspond to $|k| > \frac{N}{4}$. With non-centered indices, low frequencies correspond to $0 \leq k < \frac{N}{4}$ and $\frac{3N}{4} < k < N$ for a single set; high frequencies correspond to $\frac{N}{4} < k < \frac{3N}{4}$.

The DFT allows us to transform from the temporal (spatial) domain into the frequency domain. The **Inverse Discrete Fourier Transform, IDFT**, permits us to transform from the frequency domain back into the temporal (spatial) domain. We now give a definition for the IDFT.

Definition A.2 *Let N be an even positive integer and $\{V_m\}$ a sequence of N complex numbers where $m = -\frac{N}{2} + 1 : \frac{N}{2}$. The inverse discrete Fourier transform of $\{V_m\}$ is another sequence of N complex numbers, whose elements are given by*

$$v_j = \sum_{m=-\frac{N}{2}+1}^{\frac{N}{2}} V_m e^{\frac{i2\pi jm}{N}}, \tag{A.2}$$

for $j = -\frac{N}{2} + 1 : \frac{N}{2}$.

As with the DFT, the IDFT is defined for N odd, but we will not need this definition. Additionally, we use the operator notation $\mathcal{D}^{-1}\{V_m\}$ to denote the IDFT of the sequence $\{V_m\}$, and $\mathcal{D}^{-1}\{V_m\}_j$ to indicate the jth element of the inverse transform. $\mathcal{D}^{-1}\{V_m\}_j = v_j$.

The discrete Fourier transform and its inverse have many useful properties. Below are listed some of the more important ones that we will be using. First, however, we give the **discrete orthogonality property** for the complex exponential, which is essential in working with DFTs.

Property A.1 Orthogonality. *Let l and m be integers and let N be a positive integer. Then*

$$\sum_{j=0}^{N-1} e^{\frac{i2\pi jl}{N}} e^{-\frac{i2\pi jm}{N}} = N\hat{\delta}_N(l-m), \tag{A.3}$$

where $\hat{\delta}_N(k)$ is the **modular Kronecker delta**, *defined by*

$$\hat{\delta}_N(k) = \begin{cases} 1 & \text{if } k = 0 \text{ or a multiple of } N \\ 0 & \text{otherwise} \end{cases}.$$

Although we will not specifically need it, we include the relationship between the DFT and the IDFT:

Property A.2 Inversion. *Let $\{v_j\}$ be a sequence of N complex numbers and let $\mathcal{D}\{v_j\} = \{V_m\}$ be the DFT of this sequence. Then $\mathcal{D}^{-1}\{\mathcal{D}\{v_j\}_m\}_j = v_j$.*

The remaining properties that we will need are presented below. Proofs and detailed explanations are found in [Ref. 10].

Property A.3 Periodicity. *The complex sequences $\{v_j\}$ and $\{V_m\}$ defined by the N-point DFT pair (A.1) and (A.2) are N-periodic. That is,*

$$v_j = v_{j\pm N} \quad \text{and} \quad V_m = V_{m\pm N} \quad \text{for all intergers } j \text{ and } m.$$

Property A.4 Linearity. *If $\{v_j\}$ and $\{w_j\}$ are two complex-valued sequences of length N and α and β are two complex numbers, then*

$$\mathcal{D}\{\alpha v_j + \beta w_j\}_m = \alpha\mathcal{D}\{v_j\}_m + \beta\mathcal{D}\{w_j\}_m.$$

Property A.5 Modulation. *If the elements of the input sequence $\{v_j\}$ are multiplied by ω_N^{jk} for k a fixed integer, then the DFT of the modulated sequence is given by*

$$\mathcal{D}\{v_j\omega_N^{jk}\}_m = \mathcal{D}\{v_j\}_{m-k} = V_{m-k}.$$

Property A.6 Shifting. *If the input sequence* $\{v_j\}$ *is shifted (or translated)* k *units to the right, then*

$$\mathcal{D}\{v_{j-k}\}_m = V_m \omega_N^{-km}.$$

We make extensive use of the DFT in analyzing our solution process. The DFT is an extremely useful tool in predicting the performance of relaxation schemes (Chapter IV) and in evaluating various elements in the multilevel solution (Chapter VI) via local mode analysis. Of the properties outlined above, the orthogonality and shifting properties are the two that figure most prominently in local mode analysis.

APPENDIX B. INTERPOLATION TOOLS

Below are the interpolation tools developed by David Canright [Ref. 8] for computing the FVE stencils using *Maple*. The indexing in the following program does NOT follow the indexing in the text: the subscript i is the column indicator and corresponds to the column indicator j used in the text; the subscript j is the row indicator and corresponds to the row indicator k in the text. Thus, the indices (i, j) in this Maple program indicate the jth row and the ith column, in the text the indices (k, j) indicate the kth row and jth column.

The routine "plane" returns the function for the plane through the three given points. There is no error checking, so it has problems with special cases: all 3 points collinear (infinite solutions); and vertical planes (no solutions).

```
plane := proc(x1,y1,z1,x2,y2,z2,x3,y3,z3) local a,b,c,x,y;
unapply(
    subs(
        solve( {
            a*x1 + b*y1 + c = z1,
            a*x2 + b*y2 + c = z2,
            a*x3 + b*y3 + c = z3 },
        {a,b,c}),
    a*x + b*y + c),
x,y );
end:
```

To interpolate the indexed function z on the 6 triangles surrounding the point $(i * h, j * h)$, use the 6 functions (counter clockwise from northeast) "tri: ene, nne, nw, wsw, ssw, se". Note: these return functions of two variables (for example r, z).

```
triene := (i,j,z) - > plane(i*h,j*h,z._p,(i+1)*h,j*h,z._e,(i+1)*h,
          (j+1)*h,z._ne):
trinne := (i,j,z) - > plane(i*h,j*h,z._p,i*h,(j+1)*h,z._n,(i+1)*h,
          (j+1)*h,z._ne):
trinw := (i,j,z) - > plane(i*h,j*h,z._p,i*h,(j+1)*h,z._n,(i-1)*h,
          j*h,z._w):
triwsw := (i,j,z) - > plane(i*h,j*h,z._p,(i-1)*h,j*h,z._w,(i-1)*h,
          (j-1)*h,z._sw):
trissw := (i,j,z) - > plane(i*h,j*h,z._p,i*h,(j-1)*h,z._s,(i-1)*h,
          (j-1)*h,z._sw):
trise := (i,j,z) - > plane(i*h,j*h,z._p,i*h,(j-1)*h,z._s,(i+1)*h,
          j*h,z._e):
```

To convert the notation from subscripted back to indexed, use "toindex(expression, variable)", where the variable is to become indexed.

```
toindex := proc (expr,var) local n,m;
subs(
     zip( (x,y) - > (x=y),
          [var. (_sw,_s,_se,_w,_p,_e,_nw,_n,_ne)],
          [seq (seq (seq (var[i+n,j+m], n= -1..1),m= -1..1)]),
expr );
end:
```

Now we define shorthand for limits of integration about the general point (i, j). Here, *rdiag* means the diagonal line, where r depends on z. (This assumes integrating first in r, then in z.)

```
rp := i*h; rw := (i-1/2)*h; re := (i+1/2)*h; rdiag := i*h-(z-j*h);
zp := j*h; zs := (j-1/2)*h; zn := (j+1/2)*h;
```

Now compute the volume integral for the unknown temperature T (which is a surface integral of T after factoring out the 2π from the φ integration):

```
int (int (triene (i,j,T)(r,z)*r, r=rdiag..re),z=zp..zn) +
int (int (trinne (i,j,T)(r,z)*r, r=rp..rdiag),z=zp..zn) +
```

```
int (int (trinw (i,j,T)(r,z)*r, r=rw..rp),z=zp..zn) +
int (int (triwsw (i,j,T)(r,z)*r, r=rw..rdiag),z=zs..zp) +
int (int (trissw (i,j,T)(r,z)*r, r=rdiag..rp),z=zs..zp) +
int (int (trise (i,j,T)(r,z)*r, r=rp..re),z=zs..zp);

factor(simplify(''));
toindex('',T);
```

This set of computations produces the stencil entries for the volume integral:

$$\frac{h^3}{384}\left((32i+11)T_{i+1,j}+(32i+5)T_{i,j-1}+224iT_{i,j}+(32i-11)T_{i-1,j}\right.$$
$$\left.(16i-5)T_{i-1,j-1}+(32i-5)T_{i,j+1}+(16i+5)T_{i+1,j+1}\right).$$

In similar fashion, the surface integral of ∇T is computed (which is a circulation integral of T after factoring out the 2π from the φ integration):

```
int (+D[1] (triene(i,j,T))(re,z)*re,z=zp..zn) +
int (+D[2] (trinne(i,j,T))(r,zn)*r,r=rp..re) +
int (+D[2] (trinw(i,j,T))(r,zn)*r,r=rw..rp) +
int (-D[1] (trinw(i,j,T))(rw,z)*rw,r=zp..zn) +
int (-D[1] (triwsw(i,j,T))(rw,z)*rw,z=zs..zp) +
int (-D[2] (trissw(i,j,T))(r,zs)*r,r=rw..rp) +
int (-D[2] (trise(i,j,T))(r,zs)*r,r=rp..re) +
int (+D[1] (trise(i,j,T))(re,z)*re,z=zs..zp);

factor(simplify(''));
toindex('',T);
```

This set of computations produces the stencil entries for the surface integral:

$$\frac{h}{2}\left((1+2i)T_{i+1,j}-8iT_{i,j}+(2i-1)T_{i-1,j}+2iT_{i,j+1}+2iT_{i,j-1}\right).$$

REFERENCES

[1] Stephen F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations.* Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[2] Achi Brandt. *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics.* The von-Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, 1984.

[3] William L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[4] David Canright. Thermocapillary effects on weld pool shape, working draft. unpublished, 1994.

[5] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids, Second Edition.* Oxford University Press, Oxford, Great Britain, 1959.

[6] Eugene Isaacson and Herbert Bishop Keller. *Analysis of Numerical Methods.* Dover Publications, Mineola, New York, 1994.

[7] G. D. Smith. *Numerical Solution of Partial Differential Equations: Finite Difference Methods.* Oxford University Press, Oxford, Great Britain, 1985.

[8] David Canright. Interpolation tools, for construction of stencils for FVE method. personal communication, 1994.

[9] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1983.

[10] William L. Briggs and Van Emden Henson. *The DFT: an Owners Manual for the Discrete Fourier Transform.* Society for Industrial and Applied Mathematics, Philadelphia, to appear.

[11] C. Liu. The Finite Volume Element (FVE) and Fast Adaptive Composite Grid Methods (FAC) for the Incompressible Navier-Stokes Equations. *Copper Mountain Conference on Multigrid Methods*, 1(1):1–21, 1993.

[12] R. W. Hamming. *Numerical Methods for Scientists and Engineers, Second Edition.* Dover Publications, Mineola, New York, 1986.

[13] David Canright. Thermocapillary Flow Near a Cold Wall. *Phys. Fluids*, 6(4):1415–1424, 1994.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center — 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52 — 2
   Naval Postgraduate School
   Monterey, California 93943-5101

3. Professor Richard Franke, Code MA/Fe — 1
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5216

4. Professor David Canright, Code MA/Ca — 3
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5216

5. Professor Van Emden Henson, Code MA/Hv — 3
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5216

6. Professor Donald Danielson, Code MA/Dd — 1
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5216

7. Professor Christopher Frenzen, Code MA/Fr — 1
   Department of Mathematics
   Naval Postgraduate School
   Monterey, California 93943-5216

8. Director, Training and Education — 1
   MCCDC, Code C46
   1019 Elliot Road
   Quantico, Virginia 22134-5027

9. Requirements Division — 1
   MCCDC, Code C442
   2042 Broadway Street, Suite 11
   Quantico, Virginia 22134-5021